vSphere Storage

Update 3
VMware vSphere 8.0
VMware ESXi 8.0
vCenter Server 8.0



You can find the most up-to-date technical documentation on the VMware by Broadcom website at:

https://docs.vmware.com/

VMware by Broadcom
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About vSphere Storage

vSphere Storage describes virtualized and software-defined storage technologies that VMware ESXi™ and VMware vCenter Server® offer, and explains how to configure and use these technologies.

At VMware, we value inclusion. To foster this principle within our customer, partner, and internal community, we create content using inclusive language.

Intended Audience

This information is for experienced system administrators who are familiar with the virtual machine and storage virtualization technologies, data center operations, and SAN storage concepts.

Updated Information

This *vSphere Storage* is updated with each release of the product or when necessary.

This table provides the update history of *vSphere Storage*.

Revision	Description
14 AUG 2024	Added information about creating an NFS user account to be used by the NFS 4.1 for Kerberos authentication to Enable Kerberos Authentication in Active Directory.
29 JUL 2024	Added a required privilege to Upload Files or Folders to Datastores.
25 JUN 2024	Initial release.

Introduction to Storage in vSphere Environment

1

vSphere supports various storage options and functionalities in traditional and software-defined storage environments. A high-level overview of vSphere storage elements and aspects helps you plan a proper storage strategy for your virtual data center.

Read the following topics next:

- Traditional Storage Virtualization Models in vSphere Environment
- Software-Defined Storage Models in vSphere Environment
- vSphere Storage APIs

Traditional Storage Virtualization Models in vSphere Environment

Generally, storage virtualization refers to a logical abstraction of physical storage resources and capacities from virtual machines and their applications. ESXi provides host-level storage virtualization.

In vSphere environment, a traditional model is built around the following storage technologies and ESXi and vCenter Server virtualization functionalities.

Local and Networked Storage

In traditional storage environments, the ESXi storage management process starts with storage space that your storage administrator preallocates on different storage systems. ESXi supports local storage and networked storage.

See What Types of Physical Storage Does ESXi Support.

Storage Area Networks

A storage area network (SAN) is a specialized high-speed network that connects computer systems, or ESXi hosts, to high-performance storage systems. ESXi can use Fibre Channel or iSCSI protocols to connect to storage systems.

See Using ESXi with a SAN.

Fibre Channel

Fibre Channel (FC) is a storage protocol that the SAN uses to transfer data traffic from ESXi host servers to shared storage. The protocol packages SCSI commands into FC frames. To connect to the FC SAN, your host uses Fibre Channel host bus adapters (HBAs).

See Chapter 3 Using ESXi with Fibre Channel SAN.

Internet SCSI

Internet iSCSI (iSCSI) is a SAN transport that can use Ethernet connections between computer systems, or ESXi hosts, and high-performance storage systems. To connect to the storage systems, your hosts use hardware iSCSI adapters or software iSCSI initiators with standard network adapters.

See Chapter 4 Using ESXi with iSCSI SAN.

Storage Device or LUN

In the ESXi context, the terms storage device and LUN are used interchangeably. Typically, both terms mean a storage volume that is presented to the host from a block storage system and is available for formatting.

See Target and Device Representations and Chapter 6 Managing ESXi Storage Devices.

Virtual Disks

A virtual machine on an ESXi host uses a virtual disk to store its operating system, application files, and other data associated with its activities. Virtual disks are large physical files, or sets of files, that can be copied, moved, archived, and backed up as any other files. You can configure virtual machines with multiple virtual disks.

To access virtual disks, a virtual machine uses virtual NVMe or SCSI controllers. These virtual controllers include BusLogic Parallel, LSI Logic Parallel, LSI Logic SAS, VMware Paravirtual, NVMe, and others.

Each virtual disk resides on a datastore that is deployed on physical storage. From the standpoint of the virtual machine, each virtual disk appears as if it were a SCSI or NVMe drive connected to a SCSI or NVMe controller. Whether the physical storage is accessed through storage or network adapters on the host is typically transparent to the VM guest operating system and applications.

For information about configuring controllers for VMs, see SCSI, SATA, and NVMe Storage Controller Conditions, Limitations, and Compatibility.

VMware vSphere® VMFS

The datastores that you deploy on block storage devices use the native vSphere Virtual Machine File System (VMFS) format. It is a special high-performance file system format that is optimized for storing virtual machines.

See vSphere VMFS Datastore Concepts and Operations.

NFS

An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access an NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it as an NFS datastore.

See NFS Datastore Concepts and Operations in vSphere Environment.

Raw Device Mapping

In addition to virtual disks, vSphere offers a mechanism called raw device mapping (RDM). RDM is useful when a guest operating system inside a virtual machine requires direct access to a storage device. For information about RDMs, see Chapter 11 Raw Device Mapping in vSphere.

Software-Defined Storage Models in vSphere Environment

In addition to abstracting underlying storage capacities from vSphere VMs, as traditional storage models do, software-defined storage abstracts storage capabilities.

With the software-defined storage model, a virtual machine becomes a unit of storage provisioning and can be managed through a flexible policy-based mechanism. The model involves the following vSphere technologies.

VMware vSphere® Virtual Volumes™ (vVols)

The Virtual Volumes functionality changes the storage management paradigm from managing space inside datastores to managing abstract storage objects handled by storage arrays. With Virtual Volumes, an individual virtual machine, not the datastore, becomes a unit of storage management. And storage hardware gains complete control over virtual disk content, layout, and management.

See Chapter 14 Working with VMware vSphere Virtual Volumes.

VMware vSAN

vSAN is a distributed layer of software that runs natively as a part of the hypervisor. vSAN aggregates local or direct-attached capacity devices of an ESXi host cluster and creates a single storage pool shared across all hosts in the vSAN cluster.

See the *Administering VMware vSAN* documentation at Administering VMware vSAN.

Storage Policy Based Management

Storage Policy Based Management (SPBM) is a framework that provides a single control panel across various data services and storage solutions, including vSAN and Virtual Volumes. Using storage policies, the framework aligns application demands of your virtual machines with capabilities provided by storage entities.

See Chapter 12 Storage Policy Based Management in vSphere.

I/O Filters

I/O filters are software components that can be installed on ESXi hosts and can offer additional data services to virtual machines. Depending on implementation, the services might include replication, encryption, caching, and so on.

See Chapter 15 Filtering Virtual Machine I/O in vSphere.

vSphere Storage APIs

Storage APIs is a family of APIs used by third-party hardware, software, and storage providers to develop components that enhance several vSphere features and solutions.

This Storage publication describes several Storage APIs that contribute to your storage environment. For information about other APIs from this family, including vSphere APIs - Data Protection, see the VMware website.

vSphere APIs for Storage Awareness

Also known as VASA, these APIs, either supplied by third-party vendors or offered by VMware, enable communications between vCenter Server and underlying storage. Through VASA, storage entities can inform vCenter Server about their configurations, capabilities, and storage health and events. In return, VASA can deliver VM storage requirements from vCenter Server to a storage entity and ensure that the storage layer meets the requirements.

VASA becomes essential when you work with Virtual Volumes, vSAN, vSphere APIs for I/O Filtering (VAIO), and storage VM policies. See Chapter 13 Using Storage Providers in vSphere.

vSphere APIs for Array Integration

These APIs, also known as VAAI, include the following components:

- Hardware Acceleration APIs. Help arrays to integrate with vSphere, so that vSphere can offload certain storage operations to the array. This integration significantly reduces CPU overhead on the host. See Chapter 16 Storage Hardware Acceleration in vSphere.
- Array Thin Provisioning APIs. Help to monitor space use on thin-provisioned storage arrays to prevent out-of-space conditions, and to perform space reclamation. See ESXi and Array Thin Provisioning.

vSphere APIs for Multipathing

Known as the Pluggable Storage Architecture (PSA), these APIs allow storage partners to create and deliver multipathing and load-balancing plug-ins that are optimized for each array. Plug-ins communicate with storage arrays and determine the best path selection strategy to increase I/O performance and reliability from the ESXi host to the storage array. For more information, see Using Pluggable Storage Architecture and Path Management with ESXi.

Getting Started with a Traditional Storage Model in vSphere Environment

2

Setting up your ESXi storage in traditional environments, includes configuring your storage systems and devices, enabling storage adapters, and creating datastores.

Read the following topics next:

What Types of Physical Storage Does ESXi Support

In traditional storage environments, the ESXi storage management process starts with storage space that your storage administrator preallocates on different storage systems. ESXi supports local storage and networked storage.

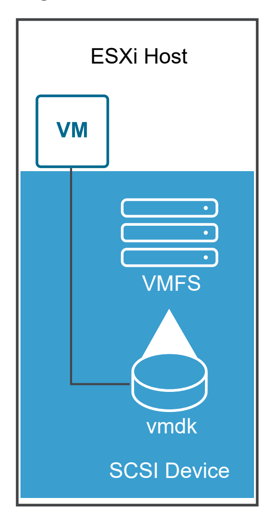
Local Storage

Local storage can be internal hard disks located inside your ESXi host. It can also include external storage systems located outside and connected to the host directly through protocols such as SAS or SATA.

Local storage does not require a storage network to communicate with your host. You need a cable connected to the storage unit and, when required, a compatible HBA in your host.

The following illustration depicts a virtual machine using local SCSI storage.

Figure 2-1. Local Storage



In this example of a local storage topology, the ESXi host uses a single connection to a storage device. On that device, you can create a VMFS datastore, which you use to store virtual machine disk files.

Although this storage configuration is possible, it is not a best practice. Using single connections between storage devices and hosts creates single points of failure (SPOF) that can cause interruptions when a connection becomes unreliable or fails. However, because most of local storage devices do not support multiple connections, you cannot use multiple paths to access local storage.

ESXi supports various local storage devices, including SCSI, IDE, SATA, USB, SAS, flash, and NVMe devices.

Note You cannot use IDE/ATA or USB drives to store virtual machines.

Local storage does not support sharing across multiple hosts. Only one host has access to a datastore on a local storage device. As a result, although you can use local storage to create VMs, you cannot use VMware features that require shared storage, such as HA and vMotion.

However, if you use a cluster of hosts that have just local storage devices, you can implement vSAN. vSAN transforms local storage resources into software-defined shared storage. With vSAN, you can use features that require shared storage. For details, see the *Administering VMware vSAN* documentation.

Networked Storage

Networked storage consists of external storage systems that your ESXi host uses to store virtual machine files remotely. Typically, the host accesses these systems over a high-speed storage network.

Networked storage devices are shared. Datastores on networked storage devices can be accessed by multiple hosts concurrently. ESXi supports multiple networked storage technologies.

In addition to traditional networked storage that this topic covers, VMware supports virtualized shared storage, such as vSAN. vSAN transforms internal storage resources of your ESXi hosts into shared storage that provides such capabilities as High Availability and vMotion for virtual machines. For details, see the *Administering VMware vSAN* documentation.

Note The same LUN cannot be presented to an ESXi host or multiple hosts through different storage protocols. To access the LUN, hosts must always use a single protocol, for example, either Fibre Channel only or iSCSI only.

Fibre Channel (FC)

Stores virtual machine files remotely on an FC storage area network (SAN). FC SAN is a specialized high-speed network that connects your hosts to high-performance storage devices. The network uses Fibre Channel protocol to transport SCSI or NVMe traffic from virtual machines to the FC SAN devices.

To connect to the FC SAN, your host should be equipped with Fibre Channel host bus adapters (HBAs). Unless you use Fibre Channel direct connect storage, you need Fibre Channel switches to route storage traffic. If your host contains FCoE (Fibre Channel over Ethernet) adapters, you can connect to your shared Fibre Channel devices by using an Ethernet network.

Fibre Channel Storage depicts virtual machines using Fibre Channel storage.

Figure 2-2. Fibre Channel Storage

In this configuration, a host connects to a SAN fabric, which consists of Fibre Channel switches and storage arrays, using a Fibre Channel adapter. LUNs from a storage array become available to the host. You can access the LUNs and create datastores for your storage needs. The datastores use the VMFS format.

For specific information on setting up the Fibre Channel SAN, see Chapter 3 Using ESXi with Fibre Channel SAN.

Internet SCSI (iSCSI)

Stores virtual machine files on remote iSCSI storage devices. iSCSI packages SCSI storage traffic into the TCP/IP protocol, so that it can travel through standard TCP/IP networks instead of the specialized FC network. With an iSCSI connection, your host serves as the initiator that communicates with a target, located in remote iSCSI storage systems.

ESXi offers the following types of iSCSI connections:

Hardware iSCSI

Your host connects to storage through a third-party adapter capable of offloading the iSCSI and network processing. Hardware adapters can be dependent and independent.

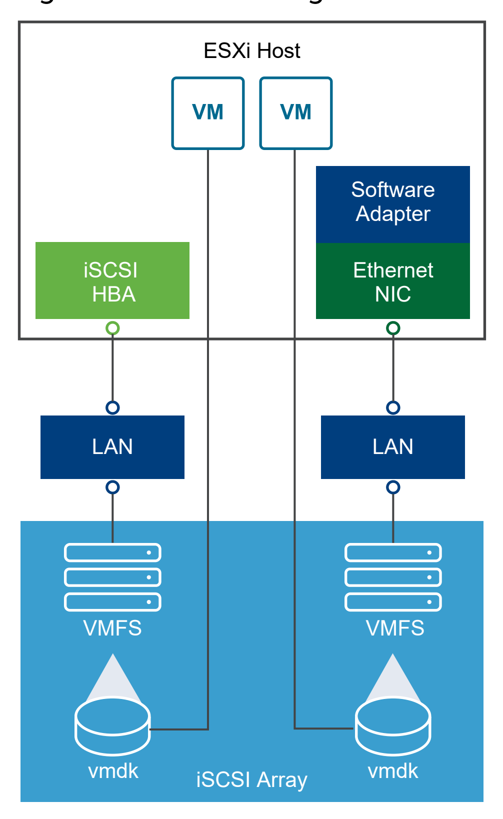
Software iSCSI

Your host uses a software-based iSCSI initiator in the VMkernel to connect to storage. With this type of iSCSI connection, your host needs only a standard network adapter for network connectivity.

You must configure iSCSI initiators for the host to access and display iSCSI storage devices.

iSCSI Storage depicts different types of iSCSI initiators.

Figure 2-3. iSCSI Storage



In the left example, the host uses the hardware iSCSI adapter to connect to the iSCSI storage system.

In the right example, the host uses a software iSCSI adapter and an Ethernet NIC to connect to the iSCSI storage.

iSCSI storage devices from the storage system become available to the host. You can access the storage devices and create VMFS datastores for your storage needs.

For specific information on setting up the iSCSI SAN, see Chapter 4 Using ESXi with iSCSI SAN.

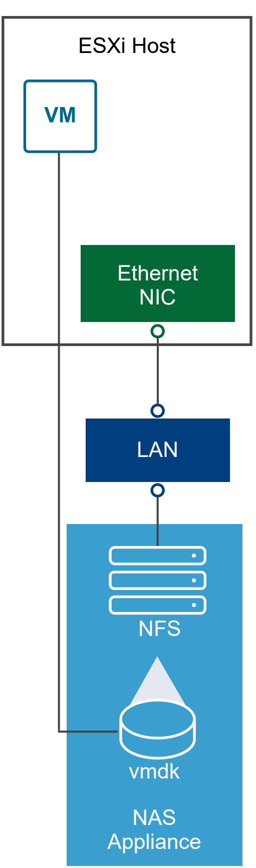
Network-attached Storage (NAS)

Stores virtual machine files on remote file servers accessed over a standard TCP/IP network. The NFS client built into ESXi uses Network File System (NFS) protocol version 3 and 4.1 to communicate with the NAS/NFS servers. For network connectivity, the host requires a standard network adapter.

You can mount an NFS volume directly on the ESXi host. You then use the NFS datastore to store and manage virtual machines in the same way that you use the VMFS datastores.

NFS Storage depicts a virtual machine using the NFS datastore to store its files. In this configuration, the host connects to the NAS server, which stores the virtual disk files, through a regular network adapter.

Figure 2-4. NFS Storage



For specific information on setting up NFS storage, see NFS Datastore Concepts and Operations in vSphere Environment.

Shared Serial Attached SCSI (SAS)

Stores virtual machines on direct-attached SAS storage systems that offer shared access to multiple hosts. This type of access permits multiple hosts to access the same VMFS datastore on a LUN.

NVMe over Fabrics Storage

VMware NVMe over Fabrics (NVMe-oF) provides a distance connectivity between a host and a target storage device on a shared storage array. VMware supports different technologies, including NVMe over RDMA (with RoCE v2 technology), NVMe over Fibre Channel, and NVMe over TCP/IP. For more information, see Chapter 8 About VMware NVMe Storage.

Comparing Types of Storage

Whether certain vSphere functionality is supported might depend on the storage technology that you use.

The following table compares networked storage technologies that ESXi supports.

Table 2-1. Networked Storage that ESXi Supports

Technology	Protocols	Transfers	Interface
Fibre Channel	FC/SCSI, FC/NVMe	Block access of data/LUN	FC HBA
Fibre Channel over Ethernet	FCoE/SCSI	Block access of data/LUN	Converged Network Adapter (hardware FCoE)
iSCSI	IP/SCSI	Block access of data/LUN	■ iSCSI HBA or iSCSI-enabled NIC (hardware iSCSI) ■ Network adapter (software iSCSI)
NAS	IP/NFS	File (no direct LUN access)	Network adapter

The following table compares the vSphere features that different types of storage support.

Table 2-2. vSphere Features Supported by Storage

Storage Type	Boot VM	vMotion	Datastore	RDM	VM Cluster	VMware HA and DRS	Storage APIs - Data Protection
Local Storage	Yes	No	VMFS	No	Yes	No	Yes
Fibre Channel	Yes	Yes	VMFS	Yes	Yes	Yes	Yes
iSCSI	Yes	Yes	VMFS	Yes	Yes	Yes	Yes
NAS over NFS	Yes	Yes	NFS 3 and NFS 4.1	No	No	Yes	Yes

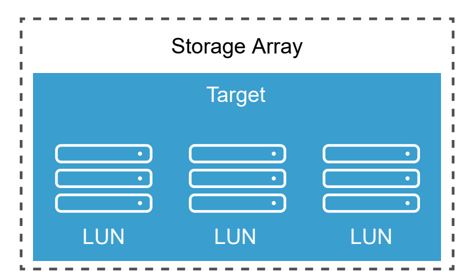
Note Local storage supports a cluster of virtual machines on a single host (also known as a cluster in a box). A shared virtual disk is required. For more information about this configuration, see the *vSphere Resource Management* documentation.

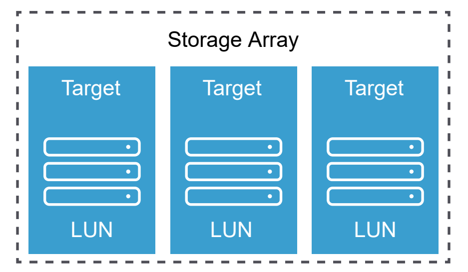
Target and Device Representations

In the ESXi context, the term target identifies a single storage unit that the host can access. The terms storage device and LUN describe a logical volume that represents storage space on a target. In the ESXi context, both terms also mean a storage volume that is presented to the host from a storage target and is available for formatting. Storage device and LUN are often used interchangeably.

Different storage vendors present the storage systems to ESXi hosts in different ways. Some vendors present a single target with multiple storage devices or LUNs on it, while others present multiple targets with one LUN each.

Figure 2-5. Target and LUN Representations





In this illustration, three LUNs are available in each configuration. In one case, the host connects to one target, but that target has three LUNs that can be used. Each LUN represents an individual storage volume. In the other example, the host detects three different targets, each having one LUN.

Targets that are accessed through the network have unique names that are provided by the storage systems. The iSCSI targets use iSCSI names. Fibre Channel targets use World Wide Names (WWNs).

Note ESXi does not support accessing the same LUN through different transport protocols, such as iSCSI and Fibre Channel.

A device, or LUN, is identified by its UUID name. If a LUN is shared by multiple hosts, it must be presented to all hosts with the same UUID.

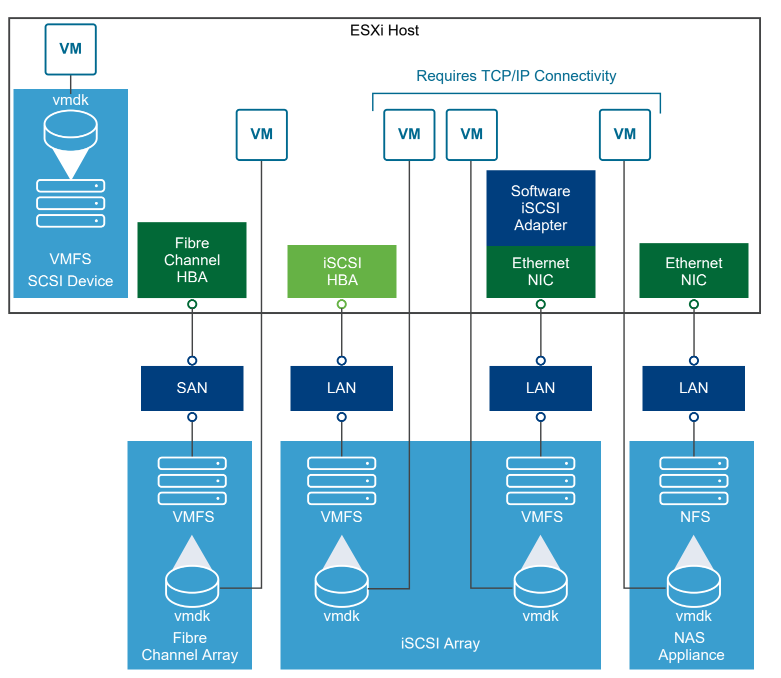
How Virtual Machines Access Storage

When a virtual machine communicates with its virtual disk stored on a datastore, it issues SCSI or NVMe commands. Because datastores can exist on various types of physical storage, these commands are encapsulated into other forms, depending on the protocol that the ESXi host uses to connect to a storage device.

Regardless of the type of storage device your host uses, the virtual disk always appears to the virtual machine as a mounted SCSI or NVMe device. The virtual disk hides a physical storage layer from the virtual machine's operating system. This allows you to run operating systems that are not certified for specific storage equipment, such as SAN, inside the virtual machine.

The following graphic depicts five virtual machines using different types of storage to illustrate the differences between each type.

Figure 2-6. Virtual machines accessing different types of storage



Note This diagram is for conceptual purposes only. It is not a recommended configuration.

Viewing Storage Devices Available to an ESXi Host

When your ESXi host connects to block-based storage systems, LUNs or storage devices that support ESXi become available to the host.

After the devices get registered with your host, you can display all available local and networked devices and review their information. If you use third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

Note If an array supports implicit asymmetric logical unit access (ALUA) and has only standby paths, the registration of the device fails. The device can register with the host after the target activates a standby path and the host detects it as active. The advanced system `/Disk/FailDiskRegistration` parameter controls this behavior of the host.

For each storage adapter, you can display a separate list of storage devices available for this adapter.

Generally, when you review storage devices, you see the following information.

Storage Device Information	Description
Name	Also called Display Name. It is a name that the ESXi host assigns to the device based on the storage type and manufacturer. Generally, you can change this name to a name of your choice. See Rename Storage Devices.
Identifier	A universally unique identifier that is intrinsic to the device. See ESXi Storage Device Names and Identifiers.

Storage Device Information	Description
Operational State	Indicates whether the device is attached or detached. See Detach Storage Devices.
LUN	Logical Unit Number (LUN) within the SCSI target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0). See Target and Device Representations .
Type	Type of device, for example, disk or CD-ROM.
Drive Type	Information about whether the device is a flash drive or a regular HDD drive. For information about flash drives and NVMe devices, see Chapter 7 Using Flash Devices with ESXi Hosts.
Transport	Transportation protocol your host uses to access the device. The protocol depends on the type of storage being used. See What Types of Physical Storage Does ESXi Support.
Capacity	Total capacity of the storage device.
Owner	The plug-in, such as the NMP or a third-party plug-in, that the host uses to manage paths to the storage device. See Using Pluggable Storage Architecture and Path Management with ESXi.
Hardware Acceleration	Information about whether the storage device assists the host with virtual machine management operations. The status can be Supported, Not Supported, or Unknown. See Chapter 16 Storage Hardware Acceleration in vSphere.
Sector Format	Indicates whether the device uses a traditional, 512n, or advanced sector format, such as 512e or 4Kn. See Device Sector Formats.
Location	A path to the storage device in the `/vmfs/devices/` directory.
Partition Format	A partition scheme used by the storage device. It can be of a master boot record (MBR) or GUID partition table (GPT) format. The GPT devices can support datastores greater than 2 TB. See Device Sector Formats.
Partitions	Primary and logical partitions, including a VMFS datastore, if configured. See Chapter 9 Working with Datastores in vSphere Storage Environment.
Multipathing Policies	Path Selection Policy and Storage Array Type Policy the host uses to manage paths to storage. See Chapter 10 Understanding Multipathing and Failover in the ESXi Environment.
Paths	Paths used to access storage and their status. See Deactivate Storage Paths.

Display Storage Devices for an ESXi Host

Display all storage devices available to a ESXi host. If you use any third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

The Storage Devices view allows you to list the hosts' storage devices, analyze their information, and modify properties.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

All storage devices available to the host are listed in the Storage Devices table.

4 To view details for a specific device, select the device from the list.

5 Use the icons to perform basic storage management tasks.

Availability of specific icons depends on the device type and configuration.

6 Use the following tabs to access additional information and modify properties for the selected device.

Tab	Description
Properties	View device properties and characteristics. View and modify multipathing policies for the device. See Viewing and Managing Storage Paths on ESXi Hosts.
Paths	Display paths available for the device. Deactivate or enable a selected path. See Deactivate Storage Paths.
Partition Details	Displays information about partitions and their formats.

Display Storage Devices for an Adapter

Display a list of storage devices accessible through a specific storage adapter on an ESXi host.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

All storage adapters installed on the host are listed in the Storage Adapters table.

4 Select the adapter from the list and click the **Devices** tab.

Storage devices that the host can access through the adapter are displayed.

5 Use the icons to perform basic storage management tasks.

Availability of specific icons depends on the device type and configuration.

Icon	Description
Refresh	Refresh information about storage adapters, topology, and file systems.
Detach Storage Devices	Detach the selected device from the host.
Attach Storage Devices	Attach the selected device to the host.
Rename Storage Devices	Change the display name of the selected device.
Use the Locator LED on ESXi Storage Devices	Turn on the locator LED for the selected devices.
Use the Locator LED on ESXi Storage Devices	Turn off the locator LED for the selected devices.

Icon	Description
Marking Storage Devices on ESXi Hosts	Mark the selected devices as flash disks.
Marking Storage Devices on ESXi Hosts	Mark the selected devices as HDD disks.
Marking Storage Devices on ESXi Hosts	Mark the selected devices as local for the host.
Marking Storage Devices on ESXi Hosts	Mark the selected devices as remote for the host.
Erase ESXi Storage Devices	Erase partitions on the selected devices.
Change Perennial Reservation Settings on ESXi Hosts	Mark the selected device as perennially reserved.
Change Perennial Reservation Settings on ESXi Hosts	Clear the perennial reservation from the selected device.

View Storage Adapters Available on an ESXi Host

Storage adapters provide connectivity for your ESXi host to a specific storage unit or network. ESXi supports different classes of adapters, including SCSI, NVMe, iSCSI, RAID, Fibre Channel, Fibre Channel over Ethernet (FCoE), and Ethernet. ESXi accesses the adapters directly through device drivers in the VMkernel.

Depending on the type of storage you use, you might need to enable and configure a storage adapter on your host.

You can display details for the available storage adapters and review their information.

Prerequisites

You must enable certain adapters, for example software iSCSI, before you can view their information. To configure adapters, see Chapter 5 Configuring iSCSI and iSER Adapters and Storage with ESXi.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

4 Use the icons to perform storage adapter tasks.

   Availability of specific icons depends on the storage configuration.

Icon	Description
Add Software Adapter	Add a storage adapter. Applies to software iSCSI.
Refresh	Refresh information about storage adapters, topology, and file systems on the host.
Rescan Storage	Rescan all storage adapters on the host to discover newly added storage devices or VMFS datastores.
Rescan Adapter	Rescan the selected adapter to discover newly added storage devices.

5 To view details for a specific adapter, select the adapter from the list.

6 Use tabs under Adapter Details to access additional information and modify properties for the selected adapter.

Tab	Description
Properties	Review general adapter properties that typically include a name and model of the adapter and unique identifiers formed according to specific storage standards. For some types of adapters, you can use this tab to configure additional properties, for example, authentication.
Devices	View storage devices the adapter can access. Use the tab to perform basic device management tasks. See Display Storage Devices for an Adapter.
Paths	List and manage all paths the adapter uses to access storage devices.
Targets (Fibre Channel and iSCSI)	Review and manage targets accessed through the adapter.
Network Port Binding (iSCSI only)	Configure port binding for software and dependent hardware iSCSI adapters.
Advanced Options (iSCSI only)	Configure advanced parameters for iSCSI.

Characteristics of ESXi Datastores

Datastores are logical containers, analogous to file systems, that hide specifics of each storage device and provide a uniform model for storing virtual machine files. You can display all datastores available to your ESXi hosts and analyze their properties.

Datastores are added to vCenter Server in the following ways:

- You can create a VMFS datastore, an NFS version 3 or 4.1 datastore, or a Virtual Volumes datastore using the New Datastore wizard. A vSAN datastore is automatically created when you enable vSAN.
- When you add an ESXi host to vCenter Server, all datastores on the host are added to vCenter Server.

The following table describes datastore details that you can see when you review datastores through the vSphere Client. Certain characteristic might not be available or applicable to all types of datastores.

Table 2-3. Datastore Information

Datastore Information	Applicable Datastore Type	Description
Name	VMFS NFS vSAN vVol	Editable name that you assign to a datastore. For information on renaming a datastore, see Change Datastore Name.
Type	VMFS NFS vSAN vVol	File system that the datastore uses. For information about VMFS and NFS datastores and how to manage them, see Chapter 9 Working with Datastores in vSphere Storage Environment. For information about vSAN datastores, see the *Administering VMware vSAN* documentation. For information about Virtual Volumes, see Chapter 14 Working with VMware vSphere Virtual Volumes.
Device Backing	VMFS NFS vSAN	Information about underlying storage, such as a storage device on which the datastore is deployed (VMFS), server and folder (NFS), or disk groups (vSAN).
Protocol Endpoints	vVol	Information about corresponding protocol endpoints. See Static Protocol Endpoints.
Extents	VMFS	Individual extents that the datastore spans and their capacity.
Drive Type	VMFS	Type of the underlying storage device, such as a flash drive or a regular HHD drive. For details, see Chapter 7 Using Flash Devices with ESXi Hosts.
Capacity	VMFS NFS vSAN vVol	Includes total capacity, provisioned space, and free space.
Mount Point	VMFS NFS vSAN vVol	A path to the datastore in the `/vmfs/volumes/` directory of the host.
Capability Sets	VMFS **Note** A multi-extent VMFS datastore assumes capabilities of only one of its extents. NFS vSAN vVol	Information about storage data services that the underlying storage entity provides. You cannot modify them.
Storage I/O Control	VMFS NFS	Information on whether cluster-wide storage I/O prioritization is enabled. See the *vSphere Resource Management* documentation.

Table 2-3. Datastore Information (continued)

Datastore Information	Applicable Datastore Type	Description
Hardware Acceleration	VMFS NFS vSAN vVol	Information on whether the underlying storage entity supports hardware acceleration. The status can be Supported, Not Supported, or Unknown. For details, see Chapter 16 Storage Hardware Acceleration in vSphere. **Note** NFS 4.1 does not support Hardware Acceleration.
Tags	VMFS NFS vSAN vVol	Datastore capabilities that you define and associate with datastores in a form of tags. For information, see Assign Tags to Datastores.
Connectivity with Hosts	VMFS NFS vVol	Hosts where the datastore is mounted.
Multipathing	VMFS vVol	Path selection policy the host uses to access storage. For more information, see Chapter 10 Understanding Multipathing and Failover in the ESXi Environment.

Display Datastore Information

Access the Datastores view with the vSphere Client navigator.

Use the Datastores view to list all datastores available in the vSphere infrastructure inventory, analyze the information, and modify properties.

Procedure

1 Navigate to any inventory object that is a valid parent object of a datastore, such as a host, a cluster, or a data center, and click the **Datastores** tab.

   Datastores that are available in the inventory appear in the center panel.

2 Use the options from a datastore right-click menu to perform basic tasks for a selected datastore.

   Availability of specific options depends on the type of the datastore and its configuration.

Option	Description
Register VM	Register an existing virtual machine in the inventory. See the *vSphere Virtual Machine Administration* documentation.
Increase Datastore Capacity	Increase the capacity of the VMFS datastore or add an extent. See Increase Capacity of vSphere VMFS Datastore .
Browse Files	Navigate to the datastore file browser. See Navigate to the Datastore Browser.
Rename	Change the datastore name. See Change Datastore Name.

Option	Description
Mount Datastore	Mount the datastore to certain hosts. See Mount Datastores.
Unmount Datastore	Unmount the datastore from certain hosts. See Unmount Datastores.
Maintenance Mode	Use datastore maintenance mode. See the *vSphere Resource Management* documentation.
Configure Storage I/O Control (VMFS)	Enable Storage I/O Control for the VMFS datastore. See the *vSphere Resource Management* documentation.
Edit Space Reclamation (VMFS)	Change space reclamation settings for the VMFS datastore. See Configure Fixed Reclamation for the Existing VMFS6 Datastore.
Delete Datastore (VMFS)	Remove the VMFS datastore. See Remove VMFS Datastores.
Tags & Custom Attributes	Use tags to encode information about the datastore. See Assign Tags to Datastores.

3 To view specific datastore details, click a selected datastore.

4 Use tabs to access additional information and modify datastore properties.

Tab	Description
Summary	View statistics and configuration for the selected datastore.
Monitor	View alarms, performance data, resource allocation, events, and other status information for the datastore.
Configure	View and modify datastore properties. Menu items that you can see depend on the datastore type.
Permissions	Assign or edit permissions for the selected datastore.
Files	Navigate to the datastore file browser.
Hosts	View hosts where the datastore is mounted.
VMs	View virtual machines that reside on the datastore.

Using Persistent Memory Devices with ESXi

ESXi supports next generation persistent memory devices, also known as Non-Volatile Memory (NVM) devices. These devices combine performance and speed of memory with the persistence of traditional storage. They can retain stored data through reboots or power source failures.

Virtual machines that require high bandwidth, low latency, and persistence can benefit from this technology. Examples include VMs with acceleration databases and analytics workload.

To use persistent memory with your ESXi host, you must be familiar with the following concepts.

PMem Datastore

After you add persistent memory to your ESXi host, the host detects the hardware, and then formats and mounts it as a local PMem datastore. ESXi uses VMFS-L as a file system format. Only one local PMem datastore per host is supported.

Note When you manage physical persistent memory, make sure to evacuate all VMs from the host and place the host into maintenance mode.

To reduce administrative overhead, the PMem datastore offers a simplified management model. Traditional datastore tasks do not generally apply to the datastore because the host automatically performs all the required operations on the background. As an administrator, you cannot display the datastore in the Datastores view of the vSphere Client, or perform other regular datastore actions. The only operation available to you is monitoring statistics for the PMem datastore.

The PMem datastore is used to store virtual NVDIMM devices and traditional virtual disks of a VM. The VM home directory with the `vmx` and `vmware.log` files cannot be placed on the PMem datastore.

PMem Access Modes

ESXi exposes persistent memory to a VM in two different modes. PMem-aware VMs can have direct access to persistent memory. Traditional VMs can use fast virtual disks stored on the PMem datastore.

Direct-Access Mode

In this mode, also called virtual PMem (vPMem) mode, a PMem region can be presented to a VM as a virtual non-volatile dual in-line memory module (NVDIMM) module. The VM uses the NVDIMM module as a standard byte-addressable memory that can persist across power cycles.

You can add one or several NVDIMM modules when provisioning the VM.

The VMs must be of the hardware version ESXi 6.7 or later and have a PMem-aware guest OS. The NVDIMM device is compatible with latest guest OSes that support persistent memory, for example, Windows 2016.

Each NVDIMM device is automatically stored on the PMem datastore.

Virtual Disk Mode

This mode, also called virtual PMem disks (vPMemDisk) mode, is available to any traditional VM and supports any hardware version, including all legacy versions. VMs are not required to be PMem-aware. When you use this mode, you create a regular SCSI virtual disk and attach a PMem VM storage policy to the disk. The policy automatically places the disk on the PMem datastore.

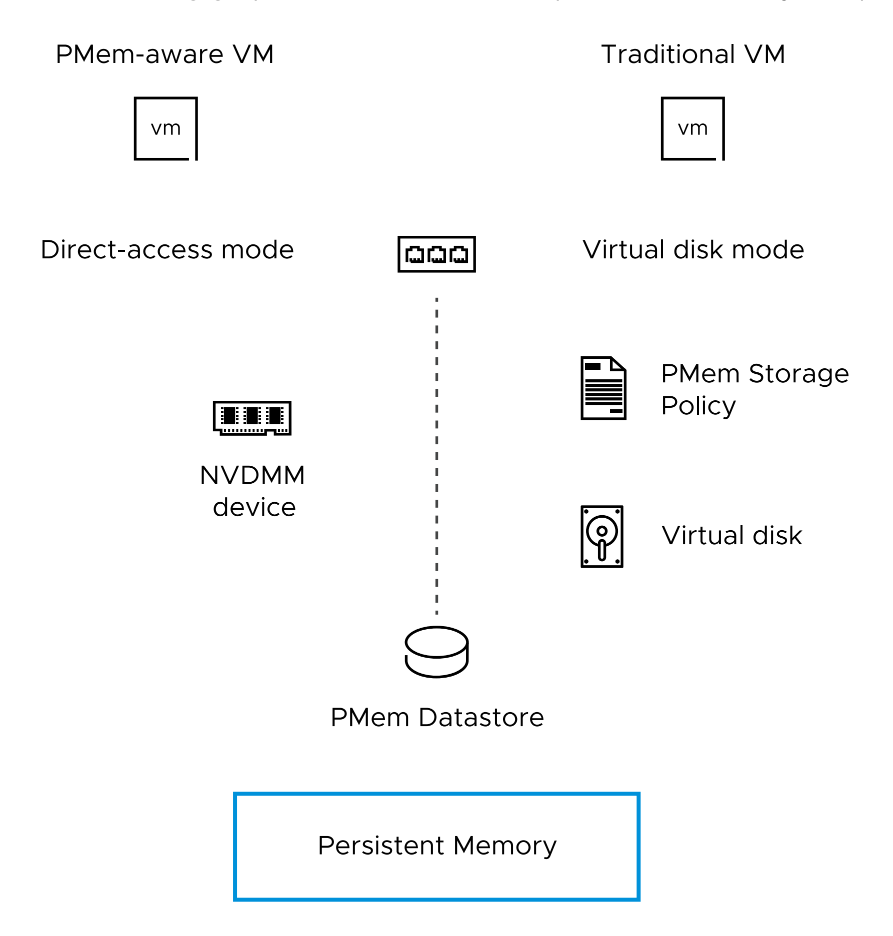
PMem Storage Policy

To place the virtual disk on the PMem datastore, you must apply the host-local PMem default storage policy to the disk. The policy is not editable.

The policy can be applied only to virtual disks. Because the VM home directory does not reside on the PMem datastore, make sure to place it on any standard datastore.

After you assign the PMem storage policy to the virtual disk, you cannot change the policy through the **VM Edit Setting** dialog box. To change the policy, migrate or clone the VM.

The following graphic illustrates how the persistent memory components interact.



For information about how to configure and manage VMs with NVDIMMs or virtual persistent memory disks, see the *vSphere Resource Management* documentation and *vSphere Virtual Machine Administration*.

Monitor PMem Datastore Statistics

You can use the vSphere Client and the `esxcli` command to review the capacity of the PMem datastore and some of its other attributes.

However, unlike regular datastores, such as VMFS or vVol, the PMem datastore does not appear in the Datastores view of the vSphere Client. Regular datastore administrative tasks do not apply to it.

Procedure

- Review PMem datastore information.

Option	Description
vSphere Client	a Navigate to the ESXi host and click **Summary**. b In the Hardware panel, verify that Persistent Memory is displayed and review its capacity.
esxcli command	Use the `esxcli storage filesystem list` to list the PMem datastore.

Example: Viewing the PMem Datastore

The following sample output appears when you use the `esxcli storage filesystem list` command to list the datastore.

```
# esxcli storage filesystem list
Mount Point             Volume Name          UUID         Mounted   Type      Size
Free
----------------------  ----------------   -------------  -------- ------- ------------
------------

/vmfs/volumes/5xxx...     ds01-102         5xxx...       true     VMFS-6  14227079168
12718178304
/vmfs/volumes/59ex...     ds02-102         59ex...       true     VMFS-6  21206401024
19697500160
/vmfs/volumes/59bx...                      59bx...       true     vfat    4293591040
4274847744
/vmfs/volumes/pmem:5ax... PMemDS-56ax...   pmem:5a0x...  true     PMEM    12880707584
11504975872
```

Using ESXi with a SAN

Using ESXi with a SAN improves flexibility, efficiency, and reliability. Using ESXi with a SAN also supports centralized management, failover, and load balancing technologies.

The following are benefits of using ESXi with a SAN:

- You can store data securely and configure multiple paths to your storage, eliminating a single point of failure.
- An ESXi host can access storage devices presented from multiple storage arrays, including arrays from different vendors.
- Using a SAN with ESXi systems extends failure resistance to the server. When you use SAN storage, all applications can instantly be restarted on another host after the failure of the original host.
- You can perform live migration of virtual machines using VMware vMotion.
- Use VMware High Availability (HA) in conjunction with a SAN to restart virtual machines in their last known state on a different server if their host fails.

- Use VMware Fault Tolerance (FT) to replicate protected virtual machines on two different hosts. Virtual machines continue to function without interruption on the secondary host if the primary one fails.
- Use VMware Distributed Resource Scheduler (DRS) to migrate virtual machines from one host to another for load balancing. Because storage is on a shared SAN array, applications continue running seamlessly.
- If you use VMware DRS clusters, put an ESXi host into maintenance mode to have the system migrate all running virtual machines to other ESXi hosts. You can then perform upgrades or other maintenance operations on the original host.

The portability and encapsulation of VMware virtual machines complements the shared nature of this storage. When virtual machines are located on SAN-based storage, you can quickly shut down a virtual machine on one server and power it up on another server, or suspend it on one server and resume operation on another server on the same network. This ability allows you to migrate computing resources while maintaining consistent shared access.

ESXi and SAN Use Cases

When used with a SAN, ESXi can benefit from multiple vSphere features, including Storage vMotion, Distributed Resource Scheduler (DRS), High Availability, and so on.

Using ESXi with a SAN is effective for the following tasks:

Storage consolidation and simplification of storage layout

If you are working with multiple hosts, and each host is running multiple virtual machines, the storage on the hosts is no longer sufficient. You might need to use external storage. The SAN can provide a simple system architecture and other benefits.

Maintenance with zero downtime

When performing ESXi host or infrastructure maintenance, use vMotion to migrate virtual machines to other host. If shared storage is on the SAN, you can perform maintenance without interruptions to the users of the virtual machines. Virtual machine working processes continue throughout a migration.

Load balancing

You can add a host to a DRS cluster, and the host's resources become part of the cluster's resources. The distribution and use of CPU and memory resources for all hosts and virtual machines in the cluster are continuously monitored. DRS compares these metrics to an ideal resource use. The ideal use considers the attributes of the cluster's resource pools and virtual machines, the current demand, and the imbalance target. If needed, DRS performs or recommends virtual machine migrations.

Disaster recovery

You can use VMware High Availability to configure multiple ESXi hosts as a cluster. The cluster provides rapid recovery from outages and cost-effective high availability for applications running in virtual machines.

Simplified array migrations and storage upgrades

When you purchase new storage systems, use Storage vMotion to perform live migrations of virtual machines from existing storage to their new destinations. You can perform the migrations without interruptions of the virtual machines.

Specifics of Using SAN Storage with ESXi

Using a SAN with an ESXi host differs from a traditional SAN use in several ways.

- You cannot use SAN administration tools to access operating systems of virtual machines that reside on the storage. With traditional tools, you can monitor only the VMware ESXi operating system. You use the vSphere Client to monitor virtual machines.
- The HBA visible to the SAN administration tools is part of the ESXi system, not part of the virtual machine.
- Typically, your ESXi system performs multipathing for you.

Making LUN Decisions

You must plan how to set up storage for your ESXi systems before you format LUNs with VMFS datastores.

When you make your LUN decision, the following considerations apply:

- Each LUN must have the correct RAID level and storage characteristic for the applications running in virtual machines that use the LUN.
- Each LUN must contain only one VMFS datastore.
- If multiple virtual machines access the same VMFS, use disk shares to prioritize virtual machines.

You might want fewer, larger LUNs for the following reasons:

- More flexibility to create virtual machines without asking the storage administrator for more space.
- More flexibility for resizing virtual disks, doing snapshots, and so on.
- Fewer VMFS datastores to manage.

You might want more, smaller LUNs for the following reasons:

- Less wasted storage space.
- Different applications might need different RAID characteristics.
- More flexibility, as the multipathing policy and disk shares are set per LUN.
- Use of Microsoft Cluster Service requires that each cluster disk resource is in its own LUN.

- Better performance because there is less contention for a single volume.

When the storage characterization for a virtual machine is unavailable, it might not be easy to determine the number and size of LUNs to provision. You can experiment using either a predictive or adaptive scheme.

Use the Predictive Scheme to Make LUN Decisions

Experiment using the predictive scheme.

Procedure

1. Provision several LUNs with different storage characteristics.
2. Create a VMFS datastore on each LUN, labeling each datastore according to its characteristics.
3. Create virtual disks to contain the data for virtual machine applications in the VMFS datastores created on LUNs with the appropriate RAID level for the applications' requirements.
4. Use disk shares to distinguish high-priority from low-priority virtual machines.

   Note Disk shares are relevant only within a given host. The shares assigned to virtual machines on one host have no effect on virtual machines on other hosts.

5. Run the applications to determine whether virtual machine performance is acceptable.

Use the Adaptive Scheme to Make LUN Decisions

You can experiment using the adaptive scheme.

Procedure

1. Provision a large LUN (RAID 1+0 or RAID 5), with write caching enabled.
2. Create a VMFS on that LUN.
3. Create four or five virtual disks on the VMFS.
4. Run the applications to determine whether disk performance is acceptable.

Results

If performance is acceptable, you can place additional virtual disks on the VMFS. If performance is not acceptable, create a new, large LUN, possibly with a different RAID level, and repeat the process. Use migration so that you do not lose virtual machines data when you recreate the LUN.

Selecting Virtual Machine Locations

When you are working on optimizing performance for your virtual machines, storage location is an important factor. Depending on your storage needs, you might select storage with high performance and high availability, or storage with lower performance.

Storage can be divided into different tiers depending on several factors:

- High Tier. Offers high performance and high availability. Might offer built-in snapshots to facilitate backups and point-in-time (PiT) restorations. Supports replication, full storage processor redundancy, and SAS drives. Uses high-cost spindles.
- Mid Tier. Offers mid-range performance, lower availability, some storage processor redundancy, and SCSI or SAS drives. Might offer snapshots. Uses medium-cost spindles.
- Lower Tier. Offers low performance, little internal storage redundancy. Uses low-end SCSI drives or SATA.

Not all VMs must be on the highest-performance and most-available storage throughout their entire life cycle.

When you decide where to place a virtual machine, the following considerations apply:

- Criticality of the VM
- Performance and availability requirements
- PiT restoration requirements
- Backup and replication requirements

A virtual machine might change tiers throughout its life cycle because of changes in criticality or changes in technology. Criticality is relative and might change for various reasons, including changes in the organization, operational processes, regulatory requirements, disaster planning, and so on.

Third-Party Management Applications

You can use third-party management applications with your ESXi host.

Most SAN hardware is packaged with storage management software. In many cases, this software is a Web application that can be used with any Web browser connected to your network. In other cases, this software typically runs on the storage system or on a single server, independent of the servers that use the SAN for storage.

Use this third-party management software for the following tasks:

- Storage array management, including LUN creation, array cache management, LUN mapping, and LUN security.
- Setting up replication, check points, snapshots, or mirroring.

If you run the SAN management software on a virtual machine, you gain the benefits of a virtual machine, including failover with vMotion and VMware HA. Because of the additional level of indirection, however, the management software might not see the SAN. In this case, you can use an RDM.

Note Whether a virtual machine can run management software successfully depends on the particular storage system.

SAN Storage Backup Considerations

Having a proper backup strategy is one of the most important aspects of SAN management. In the SAN environment, backups have two goals. The first goal is to archive online data to offline media. This process is repeated periodically for all online data on a time schedule. The second goal is to provide access to offline data for recovery from a problem. For example, database recovery often requires a retrieval of archived log files that are not currently online.

Scheduling a backup depends on several factors:

- Identification of critical applications that require more frequent backup cycles within a given period.
- Recovery point and recovery time goals. Consider how precise your recovery point must be, and how long you are willing to wait for it.
- The rate of change (RoC) associated with the data. For example, if you are using synchronous/asynchronous replication, the RoC affects the amount of bandwidth required between the primary and secondary storage devices.
- Overall impact on a SAN environment, storage performance, and other applications.
- Identification of peak traffic periods on the SAN. Backups scheduled during those peak periods can slow the applications and the backup process.
- Time to schedule all backups within the data center.
- Time it takes to back up an individual application.
- Resource availability for archiving data, such as offline media access.

Include a recovery-time objective for each application when you design your backup strategy. That is, consider the time and resources necessary to perform a backup. For example, if a scheduled backup stores so much data that recovery requires a considerable amount of time, examine the scheduled backup. Perform the backup more frequently, so that less data is backed up at a time and the recovery time decreases.

If an application requires recovery within a certain time frame, the backup process must provide a time schedule and specific data processing to meet the requirement. Fast recovery can require the use of recovery volumes that reside on online storage. This process helps to minimize or eliminate the need to access slow offline media for missing data components.

Using Third-Party Backup Packages

You can use third-party backup solutions to protect system, application, and user data in your virtual machines.

The Storage APIs - Data Protection that VMware offers can work with third-party products. When using the APIs, third-party software can perform backups without loading ESXi hosts with the processing of backup tasks.

The third-party products using the Storage APIs - Data Protection can perform the following backup tasks:

- Perform a full, differential, and incremental image backup and restore of virtual machines.
- Perform a file-level backup of virtual machines that use supported Windows and Linux operating systems.
- Ensure data consistency by using Microsoft Volume Shadow Copy Services (VSS) for virtual machines that run supported Microsoft Windows operating systems.

Because the Storage APIs - Data Protection use the snapshot capabilities of VMFS, backups do not require that you stop virtual machines. These backups are nondisruptive, can be performed at any time, and do not need extended backup windows.

For information about the Storage APIs - Data Protection and integration with backup products, see the VMware KB article 1021175.

Using ESXi with Fibre Channel SAN

3

ESXi supports Fibre Channel (FC), a storage protocol that the SAN uses to transfer data traffic from hosts to shared storage. This section provides introductory information about how to use ESXi with Fibre Channel SAN. For more information, check your vendor documentation.

Fibre Channel SAN Concepts

If you are a vSphere administrator planning to set up hosts to work with SANs, you must have a working knowledge of SAN concepts. You can find information about SANs in print and on the Internet. Because this industry changes constantly, check these resources frequently.

Storage Area Network (SAN)

A storage area network (SAN) is a specialized high-speed network that connects host servers to high-performance storage subsystems. The SAN components include host bus adapters (HBAs) in the host servers, switches that help route storage traffic, cables, storage processors (SPs), and storage disk arrays.

SAN Fabric

A SAN topology with at least one switch present on the network forms a SAN fabric.

Fibre Channel (FC) Protocol

To transfer traffic from host servers to shared storage, the SAN uses the Fibre Channel (FC) protocol that packages SCSI or NVMe commands into Fibre Channel frames.

Zoning

To restrict server access to storage arrays not allocated to that server, the SAN uses zoning. Typically, zones are created for each group of servers that access a shared group of storage devices and LUNs. Zones define which HBAs can connect to which SPs. Devices outside a zone are not visible to the devices inside the zone.

Zoning has the following effects:

- Reduces the number of targets and LUNs presented to a host.
- Controls and isolates paths in a fabric.

- Can prevent non-ESXi systems from accessing a particular storage system, and from possibly destroying VMFS data.
- Can be used to separate different environments, for example, a test from a production environment.

With ESXi hosts, use a single-initiator zoning or a single-initiator-single-target zoning. The latter is a preferred zoning practice. Using the more restrictive zoning prevents problems and misconfigurations that can occur on the SAN.

For detailed instructions and best zoning practices, contact storage array or switch vendors.

LUN Masking

Zoning is similar to LUN masking, which is commonly used for permission management. LUN masking is a process that makes a LUN available to some hosts and unavailable to other hosts.

Multipathing

When transferring data between the host server and storage, the SAN uses a technique known as multipathing. Multipathing allows you to have more than one physical path from the ESXi host to a LUN on a storage system.

Path Failover

Generally, a single path from a host to a LUN consists of an HBA, switch ports, connecting cables, and the storage controller port. If any component of the path fails, the host selects another available path for I/O. The process of detecting a failed path and switching to another is called path failover.

Ports in Fibre Channel SAN

In the context of this document, a port is the connection from a device into the SAN. Each node in the SAN, such as a host, a storage device, or a fabric component has one or more ports that connect it to the SAN. Ports are identified in a number of ways.

WWPN (World Wide Port Name)

A globally unique identifier for a port that allows certain applications to access the port. The FC switches discover the WWPN of a device or host and assign a port address to the device.

Port_ID (or port address)

Within a SAN, each port has a unique port ID that serves as the FC address for the port. This unique ID enables routing of data through the SAN to that port. The FC switches assign the port ID when the device logs in to the fabric. The port ID is valid only while the device is logged on.

When N-Port ID Virtualization (NPIV) is used, a single FC HBA port (N-port) can register with the fabric by using several WWPNs. This method allows an N-port to claim multiple fabric addresses, each of which appears as a unique entity. When ESXi hosts use a SAN, these multiple, unique identifiers allow the assignment of WWNs to individual virtual machines as part of their configuration.

Fibre Channel Storage Array Types

ESXi supports different storage systems and arrays. They generally fall into these categories.

Active-active storage system

Supports access to the LUNs simultaneously through all the storage ports that are available without significant performance degradation. All the paths are active, unless a path fails.

Active-passive storage system

A system in which one storage processor is actively providing access to a given LUN. The other processors act as a backup for the LUN and can be actively providing access to other LUN I/O. I/O can be successfully sent only to an active port for a given LUN. If access through the active storage port fails, one of the passive storage processors can be activated by the servers accessing it.

Asymmetrical storage system

Supports Asymmetric Logical Unit Access (ALUA). ALUA-compliant storage systems provide different levels of access per port. With ALUA, the host can determine the states of target ports and prioritize paths. The host uses some of the active paths as primary, and uses others as secondary.

How Virtual Machines Access Data on a Fibre Channel SAN

ESXi stores a virtual machine's disk files within a VMFS datastore that resides on a SAN storage device. When virtual machine guest operating systems send SCSI or NVMe commands to their virtual disks, the SCSI or NVMe virtualization layer translates these commands to VMFS file operations.

When a virtual machine interacts with its virtual disk stored on a SAN, the following process takes place:

1 When the guest operating system in a virtual machine reads or writes to a SCSI or NVMe disk, it sends SCSI or NVMe commands to the virtual disk.

2 Device drivers in the virtual machine’s operating system communicate with the virtual SCSI or NVMe controllers.

3 The virtual SCSI or NVMe controller forwards the command to the VMkernel.

4 The VMkernel performs the following tasks.

   a Locates the appropriate virtual disk file in the VMFS volume.

   b Maps the requests for the blocks on the virtual disk to blocks on the appropriate physical device.

   c Sends the modified I/O request from the device driver in the VMkernel to the physical HBA.

5 The physical HBA performs the following tasks.

   a Packages the I/O request according to the rules of the FC protocol.

   b Transmits the request to the SAN.

6 Depending on a port the HBA uses to connect to the fabric, one of the SAN switches receives the request. The switch routes the request to the appropriate storage device.

Read the following topics next:

- Setting Up ESXi Fibre Channel Environment
- Booting ESXi from Fibre Channel SAN
- Best Practices for Fibre Channel Storage with ESXi
- Using Fibre Channel NPIV with vSphere Virtual Machines

Setting Up ESXi Fibre Channel Environment

This topic provides an overview of setup steps that you need to follow when configuring your Fibre Channel SAN environment to work with ESXi. In preparation for configuring your Fibre Channel SAN and setting up the ESXi system, review the requirements and recommendations.

ESXi Fibre Channel SAN Requirements

- Make sure that ESXi systems support the SAN storage hardware and firmware combinations you use. For an up-to-date list, see the VMware Compatibility Guide.
- Configure your system to have only one VMFS volume per LUN.
- Unless you are using diskless servers, do not set up the diagnostic partition on a SAN LUN.

  If you use diskless servers that boot from a SAN, a shared diagnostic partition is appropriate.
- Use RDMs to access raw disks. For information, see Chapter 11 Raw Device Mapping in vSphere.
- Make sure that the storage device driver specifies a large enough queue. You can set the queue depth for the physical HBA during a system setup.
- VMtools increases the default disk timeout SCSI `TimeoutValue` parameter to 180 seconds. On virtual machines without VMtools, increase the value of the SCSI `TimeoutValue` parameter. With this parameter set up, the VMs can better tolerate delayed I/O resulting from a path failover. For information, see Path Failover and Virtual Machines or the KB 1009465.

ESXi Fibre Channel SAN Restrictions

When you use ESXi with a SAN, certain restrictions apply.

- ESXi does not support FC connected tape devices.
- You cannot use multipathing software inside a virtual machine to perform I/O load balancing to a single physical LUN. However, when your Microsoft Windows virtual machine uses dynamic disks, this restriction does not apply. For information about configuring dynamic disks, see Set Up Dynamic Disk Mirroring for Virtual Machines on vSphere VMFS.

Setting LUN Allocations

This topic provides general information about how to allocate LUNs when your ESXi works with SAN.

When you set LUN allocations, be aware of the following points:

Storage provisioning

To ensure that the ESXi system recognizes the LUNs at startup time, provision all LUNs to the appropriate HBAs before you connect the SAN to the ESXi system.

Provision all LUNs to all ESXi HBAs at the same time. HBA failover works only if all HBAs see the same LUNs.

vMotion and VMware DRS

When you use vCenter Server and vMotion or DRS, make sure that the LUNs for the virtual machines are provisioned to all ESXi hosts. This action provides the most ability to move virtual machines.

Active-active compared to active-passive arrays

When you use vMotion or DRS with an active-passive SAN storage device, make sure that all ESXi systems have consistent paths to all storage processors. Not doing so can cause path thrashing when a vMotion migration occurs.

For active-passive storage arrays not listed in Storage/SAN Compatibility, VMware does not support storage port failover. In those cases, you must connect the server to the active port on the storage array. This configuration ensures that the LUNs are presented to the ESXi host.

Setting Fibre Channel HBAs

Typically, FC HBAs that you use on your ESXi host work correctly with the default configuration settings.

You should follow the configuration guidelines provided by your storage array vendor. During FC HBA setup, consider the following issues.

- Do not mix FC HBAs from different vendors in a single host. Having different models of the same HBA is supported, but a single LUN cannot be accessed through two different HBA types, only through the same type.
- Ensure that the firmware level on each HBA is the same.
- Set the timeout value for detecting a failover. To ensure optimal performance, do not change the default value.
- ESXi supports 32 Gbps end-to-end Fibre Channel connectivity.

Configuring Fibre Channel over Ethernet

To access Fibre Channel storage, an ESXi host can use the Fibre Channel over Ethernet (FCoE) protocol. The FCoE protocol encapsulates Fibre Channel frames into Ethernet frames. As a result, your host does not need special Fibre Channel links to connect to Fibre Channel storage. The host can use 10 Gbit lossless Ethernet to deliver Fibre Channel traffic.

Fibre Channel over Ethernet Adapters

VMware supports hardware FCoE adapters, a specialized offloaded Converged Network Adapters (CNAs) that contain network and Fibre Channel functionalities on the same card.

When such adapter is installed, your host detects and can use both CNA components. In the vSphere Client, the networking component appears as a standard network adapter (vmnic) and the Fibre Channel component as a FCoE adapter (vmhba).

For information on adapters that can be used with VMware FCoE, see the *VMware Compatibility Guide*

Setup Workflow

This topic provides an overview of installation and setup steps that you need to follow when configuring your Fibre Channel SAN environment to work with ESXi.

Follow these steps to configure your ESXi SAN environment.

1. Design your SAN if it is not already configured. Most existing SANs require only minor modification to work with ESXi.
2. Check that all SAN components meet requirements.
3. Perform any necessary storage array modification.

   Most vendors have vendor-specific documentation for setting up a SAN to work with VMware ESXi.
4. Set up the HBAs for the hosts you have connected to the SAN.
5. Install ESXi on the hosts.

6 Create virtual machines and install guest operating systems.

7 (Optional) Set up your system for VMware HA failover or for using Microsoft Clustering Services.

8 Upgrade or modify your environment as needed.

Booting ESXi from Fibre Channel SAN

When you set up your ESXi host to boot from a SAN, your host's boot image is stored on one or more LUNs in the SAN storage system. When the host starts, it boots from the LUN on the SAN rather than from its local disk.

ESXi supports booting through a Fibre Channel host bus adapter (HBA) or a Fibre Channel over Ethernet (FCoE) converged network adapter (CNA).

Boot from SAN Benefits

Boot from SAN can provide numerous benefits to your ESXi environment. However, in certain cases, boot from SAN is not compatible with your hosts. Before you set up your system for boot from SAN, decide whether it is appropriate for your environment.

Caution When you use boot from SAN with multiple ESXi hosts, each host must have its own boot LUN. If you configure multiple hosts to share the boot LUN, ESXi image corruption might occur.

If you use boot from SAN, the benefits for your environment include the following:

- Cheaper servers. Servers can be more dense and run cooler without internal storage.
- Easier server replacement. You can replace servers and have the new server point to the old boot location.
- Less wasted space. Servers without local disks often take up less space.
- Easier backup processes. You can back up the system boot images in the SAN as part of the overall SAN backup procedures. Also, you can use advanced array features such as snapshots on the boot image.
- Improved management. Creating and managing the operating system image is easier and more efficient.
- Better reliability. You can access the boot disk through multiple paths, which protects the disk from being a single point of failure.

Requirements and Considerations when Booting from Fibre Channel SAN

Your ESXi boot configuration must meet specific requirements.

Table 3-1. Boot from SAN Requirements

Requirement	Description
ESXi system requirements	Follow vendor recommendations for the server booting from a SAN.
Adapter requirements	Configure the adapter, so it can access the boot LUN. See your vendor documentation.
Access control	■ Each host must have access to its own boot LUN only, not the boot LUNs of other hosts. Use storage system software to make sure that the host accesses only the designated LUNs. ■ Multiple servers can share a diagnostic partition. You can use array-specific LUN masking to achieve this configuration.
Multipathing support	Multipathing to a boot LUN on active-passive arrays is not supported because the BIOS does not support multipathing and is unable to activate a standby path.
SAN considerations	If the array is not certified for a direct connect topology, the SAN connections must be through a switched topology. If the array is certified for the direct connect topology, the SAN connections can be made directly to the array. Boot from SAN is supported for both switched topology and direct connect topology.
Hardware-specific considerations	If you are running an IBM eServer BladeCenter and use boot from SAN, you must deactivate IDE drives on the blades.

Getting Ready to Boot ESXi from SAN

When you prepare your ESXi host to boot from a SAN, you perform several tasks.

This workflow describes the generic boot-from-SAN enablement process on the rack-mounted servers. For information on enabling the boot from SAN option on blade servers or other types of hosts, refer to partner documentation.

For more information, see https://www.vmware.com/resources/compatibility/search.php.

Configure SAN Components and Storage System

Before you set up your ESXi host to boot from a SAN LUN, configure SAN components and a storage system.

Because configuring the SAN components is vendor-specific, refer to the product documentation for each item.

Procedure

1 Connect network cable, referring to any cabling guide that applies to your setup.

   Check the switch wiring, if there is any.

2 Configure the storage array.

   a From the SAN storage array, make the ESXi host visible to the SAN. This process is often called creating an object.

   b From the SAN storage array, set up the host to have the WWPNs of the host's adapters as port names or node names.

   c Create and assign LUNs.

   d Record the IP addresses of the switches and storage arrays.

   e Record the WWPN for each SP.

Caution If you use a scripted installation process to install ESXi in boot from SAN mode, take special steps to avoid unintended data loss.

Configure Storage Adapter to Boot from SAN

When you set up your host to boot from SAN, you enable the boot adapter in the host BIOS. You then configure the boot adapter to initiate a primitive connection to the target boot LUN.

Prerequisites

Determine the WWPN for the storage adapter.

Procedure

◆ Configure the storage adapter to boot from SAN.

Because configuring boot adapters is vendor-specific, consult your vendor documentation.

Set Up Your System to Boot from Installation Media

When setting up your host to boot from SAN, you first boot the host from the VMware installation media. To boot from the installation media, change the system boot sequence in the BIOS setup.

Because changing the boot sequence in the BIOS is vendor-specific, refer to vendor documentation for instructions. The following procedure explains how to change the boot sequence on an IBM host.

Procedure

1 Power on your system and enter the system BIOS Configuration/Setup Utility.

2 Select **Startup Options** and press Enter.

3 Select **Startup Sequence Options** and press Enter.

4 Change the **First Startup Device** to **[CD-ROM]**.

Results

You can now install ESXi.

Best Practices for Fibre Channel Storage with ESXi

When you use ESXi with Fibre Channel SAN, follow recommendations to avoid performance problems.

The vSphere Client offers extensive facilities for collecting performance information. The information is graphically displayed and frequently updated.

You can also use the `resxtop` or `esxtop` command-line utilities. The utilities provide a detailed look at how ESXi uses resources. For more information, see the *vSphere Resource Management* documentation.

Check with your storage representative if your storage system supports Storage API - Array Integration hardware acceleration features. If it does, refer to your vendor documentation to enable hardware acceleration support on the storage system side. For more information, see Chapter 16 Storage Hardware Acceleration in vSphere.

Preventing Fibre Channel SAN Problems

When you use ESXi with a Fibre Channel SAN, follow specific guidelines to avoid SAN problems.

To prevent problems with your SAN configuration, observe these tips:

- Place only one VMFS datastore on each LUN.
- Do not change the path policy the system sets for you unless you understand the implications of making such a change.
- Document everything. Include information about zoning, access control, storage, switch, server and FC HBA configuration, software and firmware versions, and storage cable plan.
- Plan for failure:
  - Make several copies of your topology maps. For each element, consider what happens to your SAN if the element fails.
  - Verify different links, switches, HBAs, and other elements to ensure that you did not miss a critical failure point in your design.
- Ensure that the Fibre Channel HBAs are installed in the correct slots in the host, based on slot and bus speed. Balance PCI bus load among the available buses in the server.
- Become familiar with the various monitor points in your storage network, at all visibility points, including host's performance charts, FC switch statistics, and storage performance statistics.
- Be cautious when changing IDs of the LUNs that have VMFS datastores being used by your ESXi host. If you change the ID, the datastore becomes inactive and its virtual machines fail. Resignature the datastore to make it active again. See vSphere VMFS Datastore Copies and Datastore Resignaturing.

  After you change the ID of the LUN, rescan the storage to reset the ID on your host. For information on using the rescan, see Rescan Operations for ESXi Storage.

Deactivate Automatic ESXi Host Registration

Certain storage arrays require that ESXi hosts register with the arrays. ESXi performs automatic host registration by sending the host's name and IP address to the array. If you prefer to perform manual registration using storage management software, deactivate the ESXi auto-registration feature.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **System**, click **Advanced System Settings**.
4. Under Advanced System Settings, select the **Disk.EnableNaviReg** parameter and click the **Edit** icon.
5. Change the value to 0.

Results

This operation deactivates the automatic host registration that is activated by default.

Optimizing Fibre Channel SAN Storage Performance

Several factors contribute to optimizing a typical SAN environment.

If the environment is properly configured, the SAN fabric components (particularly the SAN switches) are only minor contributors because of their low latencies relative to servers and storage arrays. Make sure that the paths through the switch fabric are not saturated, that is, that the switch fabric is running at the highest throughput.

Storage Array Performance

Storage array performance is one of the major factors contributing to the performance of the entire SAN environment.

If you encounter any problems with storage array performance, consult your storage array vendor documentation for any relevant information.

To improve the array performance in the vSphere environment, follow these general guidelines:

- When assigning LUNs, remember that several hosts might access the LUN, and that several virtual machines can run on each host. One LUN used by a host can service I/O from many different applications running on different operating systems. Because of this diverse workload, the RAID group containing the ESXi LUNs typically does not include LUNs used by other servers that are not running ESXi.
- Make sure that the read/write caching is available.

- SAN storage arrays require continual redesign and tuning to ensure that I/O is load-balanced across all storage array paths. To meet this requirement, distribute the paths to the LUNs among all the SPs to provide optimal load-balancing. Close monitoring indicates when it is necessary to rebalance the LUN distribution.

  Tuning statically balanced storage arrays is a matter of monitoring the specific performance statistics, such as I/O operations per second, blocks per second, and response time. Distributing the LUN workload to spread the workload across all the SPs is also important.

  Note Dynamic load-balancing is not currently supported with ESXi.

Server Performance with Fibre Channel

You must consider several factors to ensure optimal server performance.

Each server application must have access to its designated storage with the following conditions:

- High I/O rate (number of I/O operations per second)
- High throughput (megabytes per second)
- Minimal latency (response times)

Because each application has different requirements, you can meet these goals by selecting an appropriate RAID group on the storage array.

To achieve performance goals, follow these guidelines:

- Place each LUN on a RAID group that provides the necessary performance levels. Monitor the activities and resource use of other LUNs in the assigned RAID group. A high-performance RAID group that has too many applications doing I/O to it might not meet performance goals required by an application running on the ESXi host.
- Ensure that each host has enough HBAs to increase throughput for the applications on the host for the peak period. I/O spread across multiple HBAs provides faster throughput and less latency for each application.
- To provide redundancy for a potential HBA failure, make sure that the host is connected to a dual redundant fabric.
- When allocating LUNs or RAID groups for ESXi systems, remember that multiple operating systems use and share that resource. The LUN performance required by the ESXi host might be much higher than when you use regular physical machines. For example, if you expect to run four I/O intensive applications, allocate four times the performance capacity for the ESXi LUNs.
- When you use multiple ESXi systems in with vCenter Server, the performance requirements for the storage subsystem increase correspondingly.
- The number of outstanding I/Os needed by applications running on the ESXi system must match the number of I/Os the HBA and storage array can handle.

Using Fibre Channel NPIV with vSphere Virtual Machines

Virtual N-Port ID Virtualization (NPIV) is an ANSI T11 standard that describes how a single Fibre Channel HBA port can register with the fabric using several worldwide port names (WWPNs). This allows a fabric-attached N-port to claim multiple fabric addresses. Each address appears as a unique entity on the Fibre Channel fabric. You can configure your vSphere virtual machines to use Fibre Channel NPIV.

How NPIV-Based LUN Access Works

NPIV enables a single FC HBA port to register several unique World Wide Name (WWN) identifiers with the fabric, each of which can be assigned to an individual virtual machine. When using NPIV, a SAN administrator can monitor and route storage access per a virtual machine.

Only virtual machines with RDMs can have WWN assignments, and they use these assignments for all RDM traffic.

When a virtual machine has a WWN assigned to it, the virtual machine's configuration file (.vmx) is updated to include a WWN pair. The WWN pair consists of a World Wide Port Name (WWPN) and a World Wide Node Name (WWNN). When that virtual machine is powered on, the VMkernel creates a virtual port (VPORT) on the physical HBA which is used to access the LUN. The VPORT is a virtual HBA that appears to the FC fabric as a physical HBA. As its unique identifier, the VPORT uses the WWN pair that was assigned to the virtual machine.

Each VPORT is specific to the virtual machine. The VPORT is destroyed on the host and no longer appears to the FC fabric when the virtual machine is powered off. When a virtual machine is migrated from one host to another, the VPORT closes on the first host and opens on the destination host.

When virtual machines do not have WWN assignments, they access storage LUNs with the WWNs of their host's physical HBAs.

Requirements for Using NPIV

If you plan to enable NPIV on your virtual machines, you should be aware of certain requirements.

- NPIV can be used only for virtual machines with RDM disks. Virtual machines with regular virtual disks use the WWNs of the host's physical HBAs.
- HBAs on your host must support NPIV.

  For information, see the *VMware Compatibility Guide* and refer to your vendor documentation.

  - Use HBAs of the same type. VMware does not support heterogeneous HBAs on the same host accessing the same LUNs.
  - If a host uses multiple physical HBAs as paths to the storage, zone all physical paths to the virtual machine. This is required to support multipathing even though only one path at a time will be active.

  - Make sure that physical HBAs on the host can detect all LUNs that are to be accessed by NPIV-enabled virtual machines running on that host.
- The switches in the fabric must be NPIV-aware.
- When configuring a LUN for NPIV access at the storage level, make sure that the NPIV LUN number and NPIV target ID match the physical LUN and Target ID.
- Zone the NPIV WWPNs so that they connect to all storage systems the cluster hosts can access, even if the VM does not use the storage. If you add any new storage systems to a cluster with one or more NPIV-enabled VMs, add the new zones, so the NPIV WWPNs can detect the new storage system target ports.

NPIV Capabilities and Limitations

Learn about specific capabilities and limitations of the use of NPIV with ESXi.

ESXi with NPIV supports the following items:

- NPIV supports vMotion. When you use vMotion to migrate a virtual machine it retains the assigned WWN.

  If you migrate an NPIV-enabled virtual machine to a host that does not support NPIV, VMkernel reverts to using a physical HBA to route the I/O.
- If your FC SAN environment supports concurrent I/O on the disks from an active-active array, the concurrent I/O to two different NPIV ports is also supported.

When you use ESXi with NPIV, the following limitations apply:

- Because the NPIV technology is an extension to the FC protocol, it requires an FC switch and does not work on the direct attached FC disks.
- When you clone a virtual machine or template with a WWN assigned to it, the clones do not retain the WWN.
- NPIV does not support Storage vMotion.
- Deactivating and then re-activating the NPIV capability on an FC switch while virtual machines are running can cause an FC link to fail and I/O to stop.

Configure or Modify WWN Assignments

Assign WWN settings to a virtual machine. You can later modify the WWN assignments.

You can create from 1 to 16 WWN pairs, which can be mapped to the first 1 to 16 physical FC HBAs on the host.

Typically, you do not need to change existing WWN assignments on your virtual machine. In certain circumstances, for example, when manually assigned WWNs are causing conflicts on the SAN, you might need to change or remove WWNs.

Prerequisites

- Before configuring WWN, ensure that the ESXi host can access the storage LUN access control list (ACL) configured on the array side.
- If you want to edit the existing WWNs, power off the virtual machine.

Procedure

1. Right-click the virtual machine in the inventory and select **Edit Settings**.
2. Click **VM Options** and expand **Fibre Channel NPIV**.
3. Create or edit the WWN assignments by selecting one of the following options:

Option	Description
Temporarily disable NPIV for this virtual machine	Deactivate but do not remove the existing WWN assignments for the virtual machine.
Leave unchanged	Retain the existing WWN assignments. The read-only WWN assignments section displays the node and port values of any existing WWN assignments.
Generate new WWNs	Generate new WWNs, overwriting any existing WWNs. The WWNs of the HBA are not affected. Specify the number of WWNNs and WWPNs. A minimum of two WWPNs are required to support failover with NPIV. Typically only one WWNN is created for each virtual machine.
Remove WWN assignment	Remove the WWNs assigned to the virtual machine. The virtual machine uses the HBA WWNs to access the storage LUN.

4. Click **OK** to save your changes.

What to do next

Register newly created WWNs in the fabric.

Using ESXi with iSCSI SAN

4

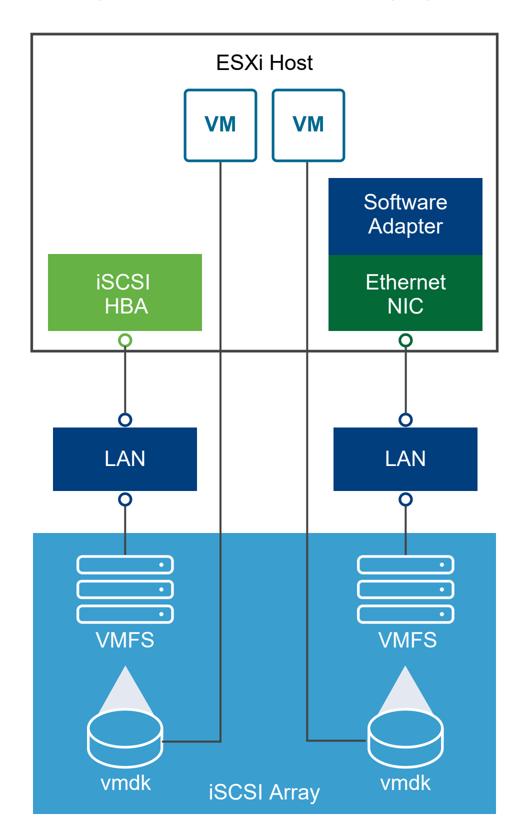
ESXi can connect to external SAN storage using the Internet SCSI (iSCSI) protocol. In addition to traditional iSCSI, ESXi also supports iSCSI Extensions for RDMA (iSER).

With the iSER protocol, the host can use the same iSCSI framework, but replaces the TCP/IP transport with the Remote Direct Memory Access (RDMA) transport.

About iSCSI SAN

iSCSI SAN uses Ethernet connections between hosts and high-performance storage subsystems.

On the host side, the iSCSI SAN components include iSCSI host bus adapters (HBAs) or Network Interface Cards (NICs). The iSCSI network also includes switches and routers that transport the storage traffic, cables, storage processors (SPs), and storage disk systems.



The iSCSI SAN uses a client-server architecture.

The client, called iSCSI initiator, operates on your ESXi host. It initiates iSCSI sessions by issuing SCSI commands and transmitting them, encapsulated into the iSCSI protocol, to an iSCSI server. The server is known as an iSCSI target. Typically, the iSCSI target represents a physical storage system on the network.

The target can also be a virtual iSCSI SAN, for example, an iSCSI target emulator running in a virtual machine. The iSCSI target responds to the initiator's commands by transmitting required iSCSI data.

iSCSI Multipathing

When transferring data between the host server and storage, the SAN uses a technique known as multipathing. With multipathing, your ESXi host can have more than one physical path to a LUN on a storage system.

Generally, a single path from a host to a LUN consists of an iSCSI adapter or NIC, switch ports, connecting cables, and the storage controller port. If any component of the path fails, the host selects another available path for I/O. The process of detecting a failed path and switching to another is called path failover.

For more information on multipathing, see Chapter 10 Understanding Multipathing and Failover in the ESXi Environment.

Nodes and Ports in the iSCSI SAN

A single discoverable entity on the iSCSI SAN, such as an initiator or a target, represents an iSCSI node.

Each node has a node name. ESXi uses several methods to identify a node.

IP Address

Each iSCSI node can have an IP address associated with it so that routing and switching equipment on your network can establish the connection between the host and storage. This address is like the IP address that you assign to your computer to get access to your company's network or the Internet.

iSCSI Name

A worldwide unique name for identifying the node. iSCSI uses the iSCSI Qualified Name (IQN) and Extended Unique Identifier (EUI).

By default, ESXi generates unique iSCSI names for your iSCSI initiators, for example, `iqn.1998-01.com.vmware:iscsitestox-68158ef2`. Usually, you do not have to change the default value, but if you do, make sure that the new iSCSI name you enter is worldwide unique.

iSCSI Alias

A more manageable name for an iSCSI device or port used instead of the iSCSI name. iSCSI aliases are not unique and are intended to be a friendly name to associate with a port.

Each node has one or more ports that connect it to the SAN. iSCSI ports are end-points of an iSCSI session.

iSCSI Naming Conventions

iSCSI uses a special unique name to identify an iSCSI node, either target or initiator.

iSCSI names are formatted in two different ways. The most common is the IQN format.

For more details on iSCSI naming requirements and string profiles, see RFC 3721 and RFC 3722 on the IETF website.

iSCSI Qualified Name Format

The iSCSI Qualified Name (IQN) format takes the form `iqn.yyyy-mm.naming-authority:unique` name, where:

- *yyyy-mm* is the year and month when the naming authority was established.
- *naming-authority* is the reverse syntax of the Internet domain name of the naming authority. For example, the `iscsi.vmware.com` naming authority can have the iSCSI qualified name form of `iqn.1998-01.com.vmware.iscsi`. The name indicates that the `vmware.com` domain name was registered in January of 1998, and `iscsi` is a subdomain, maintained by `vmware.com`.
- *unique name* is any name you want to use, for example, the name of your host. The naming authority must make sure that any names assigned following the colon are unique, such as:
  - `iqn.1998-01.com.vmware.iscsi:name1`
  - `iqn.1998-01.com.vmware.iscsi:name2`
  - `iqn.1998-01.com.vmware.iscsi:name999`

Enterprise Unique Identifier Format

The Enterprise Unique Identifier (EUI) format takes the form `eui.16_hex_digits`.

For example, `eui.0123456789ABCDEF`.

The 16-hexadecimal digits are text representations of a 64-bit number of an IEEE EUI (extended unique identifier) format. The top 24 bits are a company ID that IEEE registers with a particular company. The remaining 40 bits are assigned by the entity holding that company ID and must be unique.

iSCSI Initiators

To access iSCSI targets, your ESXi host uses iSCSI initiators.

The initiator is a software or hardware installed on your ESXi host. The iSCSI initiator originates communication between your host and an external iSCSI storage system and sends data to the storage system.

In the ESXi environment, iSCSI adapters configured on your host play the role of initiators. ESXi supports several types of iSCSI adapters.

For information on configuring and using iSCSI adapters, see Chapter 5 Configuring iSCSI and iSER Adapters and Storage with ESXi.

Software iSCSI Adapter

A software iSCSI adapter is a VMware code built into the VMkernel. Using the software iSCSI adapter, your host can connect to the iSCSI storage device through standard network adapters. The software iSCSI adapter handles iSCSI processing while communicating with the network adapter. With the software iSCSI adapter, you can use iSCSI technology without purchasing specialized hardware.

Hardware iSCSI Adapter

A hardware iSCSI adapter is a third-party adapter that offloads iSCSI and network processing from your host. Hardware iSCSI adapters are divided into categories.

- `Dependent Hardware iSCSI Adapter`. Depends on VMware networking, and iSCSI configuration and management interfaces provided by VMware.

  This type of adapter can be a card that presents a standard network adapter and iSCSI offload functionality for the same port. The iSCSI offload functionality depends on the host's network configuration to obtain the IP, MAC, and other parameters used for iSCSI sessions. An example of a dependent adapter is the iSCSI licensed Broadcom 5709 NIC.

- `Independent Hardware iSCSI Adapter`. Implements its own networking and iSCSI configuration and management interfaces.

  Typically, an independent hardware iSCSI adapter is a card that either presents only iSCSI offload functionality or iSCSI offload functionality and standard NIC functionality. The iSCSI offload functionality has independent configuration management that assigns the IP, MAC, and other parameters used for the iSCSI sessions. An example of an independent adapter is the QLogic QLA4052 adapter.

Hardware iSCSI adapters might need to be licensed. Otherwise, they might not appear in the client or vSphere CLI. Contact your vendor for licensing information.

Using iSER Protocol with ESXi

In addition to traditional iSCSI, ESXi supports the iSCSI Extensions for RDMA (iSER) protocol. When the iSER protocol is enabled, the iSCSI framework on the ESXi host can use the Remote Direct Memory Access (RDMA) transport instead of TCP/IP.

The traditional iSCSI protocol carries SCSI commands over a TCP/IP network between an iSCSI initiator on a host and an iSCSI target on a storage device. The iSCSI protocol encapsulates the commands and assembles that data in packets for the TCP/IP layer. When the data arrives, the iSCSI protocol disassembles the TCP/IP packets, so that the SCSI commands can be differentiated and delivered to the storage device.

iSER differs from traditional iSCSI as it replaces the TCP/IP data transfer model with the Remote Direct Memory Access (RDMA) transport. Using the direct data placement technology of the RDMA, the iSER protocol can transfer data directly between the memory buffers of the ESXi host and storage devices. This method eliminates unnecessary TCP/IP processing and data coping, and can also reduce latency and the CPU load on the storage device.

In the iSER environment, iSCSI works exactly as before, but uses an underlying RDMA fabric interface instead of the TCP/IP-based interface.

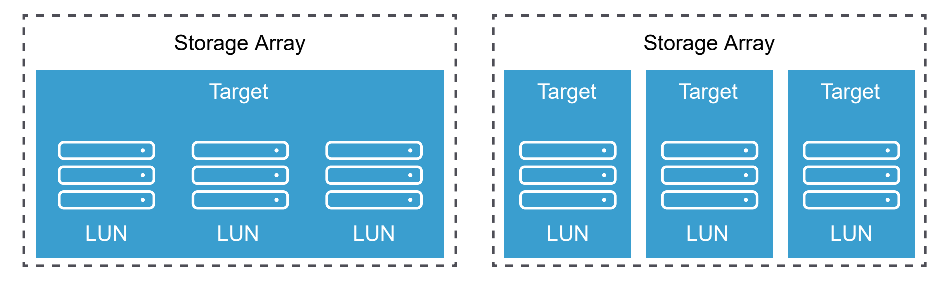
Because the iSER protocol preserves the compatibility with iSCSI infrastructure, the process of enabling iSER on the ESXi host is similar to the iSCSI process. See Configure iSER with ESXi.

Establishing iSCSI Connections

In the ESXi context, the term target identifies a single storage unit that your host can access. The terms storage device and LUN describe a logical volume that represents storage space on a target. Typically, the terms device and LUN, in the ESXi context, mean a SCSI volume presented to your host from a storage target and available for formatting.

Different iSCSI storage vendors present storage to hosts in different ways. Some vendors present multiple LUNs on a single target. Others present multiple targets with one LUN each.

Figure 4-1. Target Compared to LUN Representations



In these examples, three LUNs are available in each of these configurations. In the first case, the host detects one target but that target has three LUNs that can be used. Each of the LUNs represents individual storage volume. In the second case, the host detects three different targets, each having one LUN.

Host-based iSCSI initiators establish connections to each target. Storage systems with a single target containing multiple LUNs have traffic to all the LUNs on a single connection. With a system that has three targets with one LUN each, the host uses separate connections to the three LUNs.

This information is useful when you are trying to aggregate storage traffic on multiple connections from the host with multiple iSCSI adapters. You can set the traffic for one target to a particular adapter, and use a different adapter for the traffic to another target.

iSCSI Storage System Types

The types of ESXi storage that your host supports include active-active, active-passive, and ALUA-compliant.

Active-active storage system

Supports access to the LUNs simultaneously through all the storage ports that are available without significant performance degradation. All the paths are always active, unless a path fails.

Active-passive storage system

A system in which one storage processor is actively providing access to a given LUN. The other processors act as a backup for the LUN and can be actively providing access to other LUN I/O. I/O can be successfully sent only to an active port for a given LUN. If access through the active storage port fails, one of the passive storage processors can be activated by the servers accessing it.

Asymmetrical storage system

Supports Asymmetric Logical Unit Access (ALUA). ALUA-compliant storage systems provide different levels of access per port. With ALUA, hosts can determine the states of target ports and prioritize paths. The host uses some of the active paths as primary and others as secondary.

Virtual port storage system

Supports access to all available LUNs through a single virtual port. Virtual port storage systems are active-active storage devices, but hide their multiple connections though a single port. ESXi multipathing does not make multiple connections from a specific port to the storage by default. Some storage vendors supply session managers to establish and manage multiple connections to their storage. These storage systems handle port failovers and connection balancing transparently. This capability is often called transparent failover.

Discovery, Authentication, and Access Control

You can use several mechanisms to discover your storage and to limit access to it.

You must configure your host and the iSCSI storage system to support your storage access control policy.

Discovery

A discovery session is part of the iSCSI protocol. It returns the set of targets you can access on an iSCSI storage system. The two types of discovery available on ESXi are dynamic and static. Dynamic discovery obtains a list of accessible targets from the iSCSI storage system. Static discovery can access only a particular target by target name and address.

For more information, see Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host.

Authentication

iSCSI storage systems authenticate an initiator by a name and key pair. ESXi supports the CHAP authentication protocol. To use CHAP authentication, the ESXi host and the iSCSI storage system must have CHAP enabled and have common credentials.

For information on enabling CHAP, see Configuring CHAP Parameters for iSCSI or iSER Storage Adapters on ESXi Host.

Access Control

Access control is a policy set up on the iSCSI storage system. Most implementations support one or more of three types of access control:

- By initiator name
- By IP address
- By the CHAP protocol

Only initiators that meet all rules can access the iSCSI volume.

Using only CHAP for access control can slow down rescans because the ESXi host can discover all targets, but then fails at the authentication step. iSCSI rescans work faster if the host discovers only the targets it can authenticate.

How Virtual Machines Access Data on an iSCSI SAN

ESXi stores a virtual machine's disk files within a VMFS datastore that resides on a SAN storage device. When virtual machine guest operating systems send SCSI commands to their virtual disks, the SCSI virtualization layer translates these commands to VMFS file operations.

When a virtual machine interacts with its virtual disk stored on a SAN, the following process takes place:

1 When the guest operating system in a virtual machine reads or writes to SCSI disk, it sends SCSI commands to the virtual disk.

2 Device drivers in the virtual machine’s operating system communicate with the virtual SCSI controllers.

3 The virtual SCSI controller forwards the commands to the VMkernel.

4 The VMkernel performs the following tasks.

    a Locates an appropriate virtual disk file in the VMFS volume.

   b Maps the requests for the blocks on the virtual disk to blocks on the appropriate physical device.

   c Sends the modified I/O request from the device driver in the VMkernel to the iSCSI initiator, hardware or software.

5 If the iSCSI initiator is a hardware iSCSI adapter, independent or dependent, the adapter performs the following tasks.

   a Encapsulates I/O requests into iSCSI Protocol Data Units (PDUs).

   b Encapsulates iSCSI PDUs into TCP/IP packets.

   c Sends IP packets over Ethernet to the iSCSI storage system.

6 If the iSCSI initiator is a software iSCSI adapter, the following takes place.

   a The iSCSI initiator encapsulates I/O requests into iSCSI PDUs.

   b The initiator sends iSCSI PDUs through TCP/IP connections.

   c The VMkernel TCP/IP stack relays TCP/IP packets to a physical NIC.

   d The physical NIC sends IP packets over Ethernet to the iSCSI storage system.

7 Ethernet switches and routers on the network carry the request to the appropriate storage device.

Error Correction

To protect the integrity of iSCSI headers and data, the iSCSI protocol offers error correction methods known as header digests and data digests.

Both methods are supported by the ESXi host, and you can activate them. These methods are used to check the header and SCSI data transferred between iSCSI initiators and targets, in both directions.

Header and data digests check the noncryptographic data integrity beyond the integrity checks that other networking layers, such as TCP and Ethernet, provide. The digests check the entire communication path, including all elements that can change the network-level traffic, such as routers, switches, and proxies.

The existence and type of the digests are negotiated when an iSCSI connection is established. When the initiator and target agree on a digest configuration, this digest must be used for all traffic between them.

Enabling header and data digests requires additional processing for both the initiator and the target, and can affect throughput and CPU use performance.

Note Systems that use the Intel Nehalem processors offload the iSCSI digest calculations, as a result, reducing the impact on performance.

For information on enabling header and data digests, see Configuring Advanced Parameters for iSCSI on ESXi Host.

Read the following topics next:

- Booting from iSCSI SAN with ESXi
- Best Practices for iSCSI Storage with ESXi

Booting from iSCSI SAN with ESXi

When you set up your ESXi host to boot from a SAN, your host's boot image is stored on one or more LUNs in the SAN storage system. When the ESXi host starts, it boots from the LUN on the SAN rather than from its local disk.

You can use boot from the SAN if you do not want to handle maintenance of local storage or have diskless hardware configurations, such as blade systems.

ESXi supports different methods of booting from the iSCSI SAN.

Table 4-1. Boot from iSCSI SAN support

Independent Hardware iSCSI	Software iSCSI
Configure the iSCSI HBA to boot from the SAN. For information on configuring the HBA, see Configure Independent Hardware iSCSI Adapter for SAN Boot.	Use the software iSCSI adapter and a network adapter that supports the iSCSI Boot Firmware Table (iBFT) format. For information, see *VMware ESXi Installation and Setup*.

General Recommendations for Boot from iSCSI SAN

If you plan to set up and use an iSCSI LUN as the boot device for your host, follow certain general guidelines.

The following guidelines apply to booting from the independent hardware iSCSI and iBFT.

- Review any vendor recommendations for the hardware you use in your boot configuration.
- For installation prerequisites and requirements, review *vSphere Installation and Setup*.
- Use static IP addresses to reduce the chances of DHCP conflicts.
- Use different LUNs for VMFS datastores and boot partitions.
- Configure proper ACLs on your storage system.
  - The boot LUN must be visible only to the host that uses the LUN. No other host on the SAN is permitted to see that boot LUN.
  - If a LUN is used for a VMFS datastore, multiple hosts can share the LUN.

- Configure a diagnostic partition.
  - With the independent hardware iSCSI only, you can place the diagnostic partition on the boot LUN. If you configure the diagnostic partition in the boot LUN, this LUN cannot be shared across multiple hosts. If a separate LUN is used for the diagnostic partition, multiple hosts can share the LUN.
  - If you boot from SAN using iBFT, you cannot set up a diagnostic partition on a SAN LUN. To collect your host's diagnostic information, use the vSphere ESXi Dump Collector on a remote server. For information about the ESXi Dump Collector, see *vCenter Server Installation and Setup* and *vSphere Networking*.

Prepare the iSCSI SAN

Before you configure your host to boot from an iSCSI LUN, prepare and configure your storage area network.

Caution If you use scripted installation to install ESXi when booting from a SAN, you must take special steps to avoid unintended data loss.

Procedure

1. Connect network cables, referring to any cabling guide that applies to your setup.
2. Ensure IP connectivity between your storage system and server.

   Verify configuration of any routers or switches on your storage network. Storage systems must be able to ping the iSCSI adapters in your hosts.
3. Configure the storage system.
   a. Create a volume (or LUN) on the storage system for your host to boot from.
   b. Configure the storage system so that your host has access to the assigned LUN.

      This step might involve updating ACLs with the IP addresses, iSCSI names, and the CHAP authentication parameter you use on your host. On some storage systems, in addition to providing access information for the ESXi host, you must also explicitly associate the assigned LUN with the host.
   c. Ensure that the LUN is presented to the host correctly.
   d. Ensure that no other system has access to the configured LUN.
   e. Record the iSCSI name and IP addresses of the targets assigned to the host.

      You must have this information to configure your iSCSI adapters.

Configure Independent Hardware iSCSI Adapter for SAN Boot

If your ESXi host uses an independent hardware iSCSI adapter, such as QLogic HBA, you can configure the adapter to boot from the SAN.

This procedure discusses how to enable the QLogic iSCSI HBA to boot from the SAN. For more information and more up-to-date details about QLogic adapter configuration settings, see the QLogic website.

Procedure

1 Start the installation media and reboot the host.

2 Use the BIOS to set the host to boot from the installation media first.

3 During server POST, press Crtl+q to enter the QLogic iSCSI HBA configuration menu.

4 Select the I/O port to configure.

   By default, the Adapter Boot mode is set to Disable.

5 Configure the HBA.

   a From the **Fast!UTIL Options** menu, select **Configuration Settings > Host Adapter Settings**.

   b (Optional) Configure the following settings for your host adapter: initiator IP address, subnet mask, gateway, initiator iSCSI name, and CHAP.

6 Configure iSCSI settings.

   See Configure iSCSI Boot Settings.

7 Save your changes and restart the system.

Configure iSCSI Boot Settings

Configure iSCSI boot parameters, so that your ESXi host can boot from an iSCSI LUN.

Procedure

1 From the **Fast!UTIL Options** menu, select **Configuration Settings > iSCSI Boot Settings**.

2 Before you can set SendTargets, set Adapter Boot mode to **Manual**.

3 Select **Primary Boot Device Settings**.

   a Enter the discovery **Target IP** and **Target Port**.

   b Configure the **Boot LUN** and **iSCSI Name** parameters.

      - If only one iSCSI target and one LUN are available at the target address, leave **Boot LUN** and **iSCSI Name** blank.

        After your host reaches the target storage system, these text boxes are populated with appropriate information.

      - If more than one iSCSI target and LUN are available, supply values for **Boot LUN** and **iSCSI Name**.

   c Save changes.

4 From the **iSCSI Boot Settings** menu, select the primary boot device.

   An auto rescan of the HBA finds new target LUNs.

5 Select the iSCSI target.

   If more than one LUN exists within the target, you can select a specific LUN ID by pressing **Enter** after you locate the iSCSI device.

6 Return to the **Primary Boot Device Setting** menu. After the rescan, **Boot LUN** and **iSCSI Name** are populated. Change the value of **Boot LUN** to the appropriate LUN ID.

Best Practices for iSCSI Storage with ESXi

When you use ESXi with iSCSI SAN, follow recommendations that VMware offers to avoid problems.

Check with your storage representative if your storage system supports Storage API - Array Integration hardware acceleration features. If it does, refer to your vendor documentation to enable hardware acceleration support on the storage system side. For more information, see Chapter 16 Storage Hardware Acceleration in vSphere.

Preventing iSCSI SAN Problems

When using ESXi with a SAN, you must follow specific guidelines to avoid SAN problems.

Observe the following tips:

- Place only one VMFS datastore on each LUN.
- Do not change the path policy the system sets for you unless you understand the implications of making such a change.
- Document everything. Include information about configuration, access control, storage, switch, server and iSCSI HBA configuration, software and firmware versions, and storage cable plan.
- Plan for failure:
  - Make several copies of your topology maps. For each element, consider what happens to your SAN if the element fails.
  - Cross off different links, switches, HBAs, and other elements to ensure that you did not miss a critical failure point in your design.
- Ensure that the iSCSI HBAs are installed in the correct slots in the ESXi host, based on slot and bus speed. Balance PCI bus load among the available buses in the server.
- Become familiar with the various monitor points in your storage network, at all visibility points, including ESXi performance charts, Ethernet switch statistics, and storage performance statistics.
- Change LUN IDs only when VMFS datastores deployed on the LUNs have no running virtual machines. If you change the ID, virtual machines running on the VMFS datastore might fail.

After you change the ID of the LUN, you must rescan your storage to reset the ID on your host. For information on using the rescan, see Rescan Operations for ESXi Storage.

- If you change the default iSCSI name of your iSCSI adapter, make sure that the name you enter is worldwide unique and properly formatted. To avoid storage access problems, never assign the same iSCSI name to different adapters, even on different hosts.

Optimizing iSCSI SAN Storage Performance

Several factors contribute to optimizing a typical SAN environment.

If the network environment is properly configured, the iSCSI components provide adequate throughput and low enough latency for iSCSI initiators and targets. If the network is congested and links, switches or routers are saturated, iSCSI performance suffers and might not be adequate for ESXi environments.

Storage System Performance

Storage system performance is one of the major factors contributing to the performance of the entire iSCSI environment.

If issues occur with storage system performance, consult your storage system vendor's documentation for any relevant information.

When you assign LUNs, remember that you can access each shared LUN through a number of hosts, and that a number of virtual machines can run on each host. One LUN used by the ESXi host can service I/O from many different applications running on different operating systems. Because of this diverse workload, the RAID group that contains the ESXi LUNs should not include LUNs that other hosts use that are not running ESXi for I/O intensive applications.

Enable read caching and write caching.

Load balancing is the process of spreading server I/O requests across all available SPs and their associated host server paths. The goal is to optimize performance in terms of throughput (I/O per second, megabytes per second, or response times).

SAN storage systems require continual redesign and tuning to ensure that I/O is load balanced across all storage system paths. To meet this requirement, distribute the paths to the LUNs among all the SPs to provide optimal load balancing. Close monitoring indicates when it is necessary to manually rebalance the LUN distribution.

Tuning statically balanced storage systems is a matter of monitoring the specific performance statistics (such as I/O operations per second, blocks per second, and response time) and distributing the LUN workload to spread the workload across all the SPs.

Server Performance with iSCSI

To ensure optimal ESXi host performance, consider several factors.

Each server application must have access to its designated storage with the following conditions:

- High I/O rate (number of I/O operations per second)

- High throughput (megabytes per second)
- Minimal latency (response times)

Because each application has different requirements, you can meet these goals by selecting an appropriate RAID group on the storage system.

To achieve performance goals, follow these guidelines:

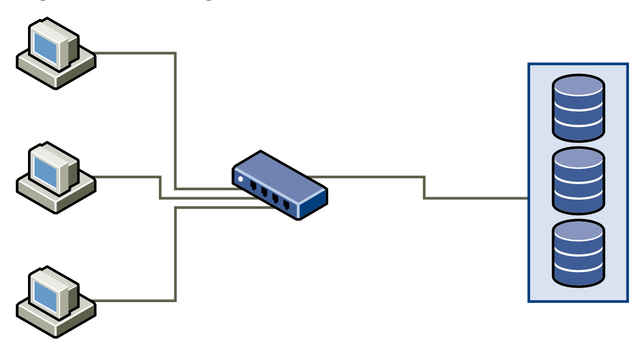
- Place each LUN on a RAID group that provides the necessary performance levels. Monitor the activities and resource use of other LUNS in the assigned RAID group. A high-performance RAID group that has too many applications doing I/O to it might not meet performance goals required by an application running on the ESXi host.
- To achieve maximum throughput for all the applications on the host during the peak period, install enough network adapters or iSCSI hardware adapters. I/O spread across multiple ports provides faster throughput and less latency for each application.
- To provide redundancy for software iSCSI, make sure that the initiator is connected to all network adapters used for iSCSI connectivity.
- When allocating LUNs or RAID groups for ESXi systems, remember that multiple operating systems use and share that resource. The LUN performance required by the ESXi host might be much higher than when you use regular physical machines. For example, if you expect to run four I/O intensive applications, allocate four times the performance capacity for the ESXi LUNs.
- When you use multiple ESXi systems with vCenter Server, the storage performance requirements increase.
- The number of outstanding I/Os needed by applications running on an ESXi system must match the number of I/Os the SAN can handle.

Network Performance

A typical SAN consists of a collection of computers connected to a collection of storage systems through a network of switches. Several computers often access the same storage.

The following graphic shows several computer systems connected to a storage system through an Ethernet switch. In this configuration, each system is connected through a single Ethernet link to the switch. The switch is connected to the storage system through a single Ethernet link.

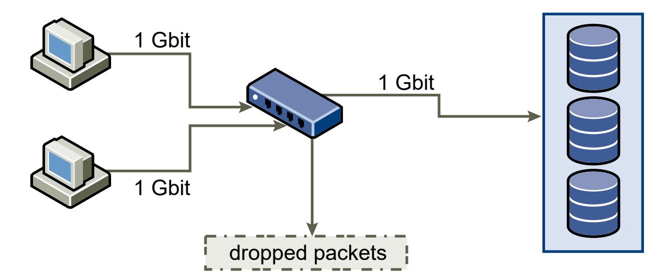
Figure 4-2. Single Ethernet Link Connection to Storage



When systems read data from storage, the storage responds with sending enough data to fill the link between the storage systems and the Ethernet switch. It is unlikely that any single system or virtual machine gets full use of the network speed. However, this situation can be expected when many systems share one storage device.

When writing data to storage, multiple systems or virtual machines might attempt to fill their links. As a result, the switch between the systems and the storage system might drop network packets. The data drop might occur because the switch has more traffic to send to the storage system than a single link can carry. The amount of data the switch can transmit is limited by the speed of the link between it and the storage system.

Figure 4-3. Dropped Packets



Recovering from dropped network packets results in large performance degradation. In addition to time spent determining that data was dropped, the retransmission uses network bandwidth that can otherwise be used for current transactions.

iSCSI traffic is carried on the network by the Transmission Control Protocol (TCP). TCP is a reliable transmission protocol that ensures that dropped packets are retried and eventually reach their destination. TCP is designed to recover from dropped packets and retransmits them quickly and seamlessly. However, when the switch discards packets with any regularity, network throughput suffers. The network becomes congested with requests to resend data and with the resent packets. Less data is transferred than in a network without congestion.

Most Ethernet switches can buffer, or store, data. This technique gives every device attempting to send data an equal chance to get to the destination. The ability to buffer some transmissions, combined with many systems limiting the number of outstanding commands, reduces transmissions to small bursts. The bursts from several systems can be sent to a storage system in turn.

If the transactions are large and multiple servers are sending data through a single switch port, an ability to buffer can be exceeded. In this case, the switch drops the data it cannot send, and the storage system must request a retransmission of the dropped packet. For example, if an Ethernet switch can buffer 32 KB, but the server sends 256 KB to the storage device, some of the data is dropped.

Most managed switches provide information on dropped packets, similar to the following:

```
*: interface is up
IHQ: pkts in input hold queue      IQD: pkts dropped from input queue
OHQ: pkts in output hold queue     OQD: pkts dropped from output queue
RXBS: rx rate (bits/sec)           RXPS: rx rate (pkts/sec)
TXBS: tx rate (bits/sec)           TXPS: tx rate (pkts/sec)
TRTL: throttle count
```

Table 4-2. Sample Switch Information

Interface	IHQ	IQD	OHQ	OQD	RXBS	RXPS	TXBS	TXPS	TRTL
* GigabitEthernet0/1	3	9922	0	0	476303000	62273	477840000	63677	0

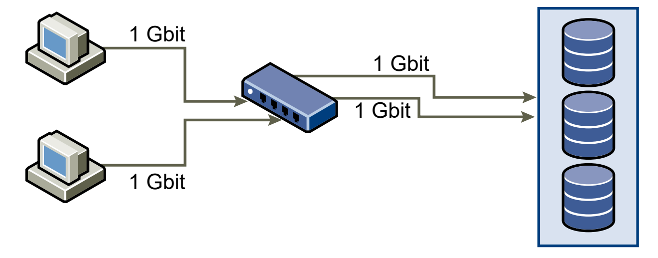
In this example from a Cisco switch, the bandwidth used is 476303000 bits/second, which is less than half of wire speed. The port is buffering incoming packets, but has dropped several packets. The final line of this interface summary indicates that this port has already dropped almost 10,000 inbound packets in the IQD column.

Configuration changes to avoid this problem involve making sure several input Ethernet links are not funneled into one output link, resulting in an oversubscribed link. When several links transmitting near capacity are switched to a smaller number of links, oversubscription becomes possible.

Generally, applications or systems that write much data to storage must avoid sharing Ethernet links to a storage device. These types of applications perform best with multiple connections to storage devices.

Multiple Connections from Switch to Storage shows multiple connections from the switch to the storage.

Figure 4-4. Multiple Connections from Switch to Storage



Using VLANs or VPNs does not provide a suitable solution to the problem of link oversubscription in shared configurations. VLANs and other virtual partitioning of a network provide a way of logically designing a network. However, they do not change the physical capabilities of links and trunks between switches. When storage traffic and other network traffic share physical connections, oversubscription and lost packets might become possible. The same is true of VLANs that share interswitch trunks. Performance design for a SAN must consider the physical limitations of the network, not logical allocations.

Checking Ethernet Switch Statistics

Many Ethernet switches provide different methods for monitoring switch health.

Switches that have ports operating near maximum throughput much of the time do not provide optimum performance. If you have ports in your iSCSI SAN running near the maximum, reduce the load. If the port is connected to an ESXi system or iSCSI storage, you can reduce the load by using manual load balancing.

If the port is connected between multiple switches or routers, consider installing additional links between these components to handle more load. Ethernet switches also commonly provide information about transmission errors, queued packets, and dropped Ethernet packets. If the switch regularly reports any of these conditions on ports being used for iSCSI traffic, performance of the iSCSI SAN will be poor.

Configuring iSCSI and iSER Adapters and Storage with ESXi

5

Before ESXi can work with iSCSI SAN, you must set up your iSCSI environment. To work properly, your environment must follow specific recommendations. In addition, several restrictions exist when you use ESXi with iSCSI SAN.

For additional information about using vSphere with iSCSI, see Best Practices For Running VMware vSphere On iSCSI blog.

iSCSI Storage Recommendations

- Verify that your ESXi host supports the iSCSI SAN storage hardware and firmware. For an up-to-date list, see *VMware Compatibility Guide*.
- To ensure that the host recognizes LUNs at startup time, configure all iSCSI storage targets so that your host can access them and use them. Configure your host so that it can discover all available iSCSI targets.
- Unless you are using diskless servers, set up a diagnostic partition on local storage. If you have diskless servers that boot from iSCSI SAN, see General Recommendations for Boot from iSCSI SAN for information about diagnostic partitions with iSCSI.
- Set the SCSI controller driver in the guest operating system to a large enough queue.
- VMtools increases the default disk timeout SCSI `TimeoutValue` parameter to 180 seconds. On virtual machines without VMtools, increase the value of the SCSI `TimeoutValue` parameter. With this parameter set up, the VMs can better tolerate delayed I/O resulting from a path failover. For information, see Path Failover and Virtual Machines or the KB 1009465.
- Configure your environment to have only one VMFS datastore for each LUN.

iSCSI Storage Restrictions

- ESXi does not support iSCSI-connected tape devices.
- You cannot use virtual-machine multipathing software to perform I/O load balancing to a single physical LUN.
- ESXi does not support multipathing when you combine independent hardware adapters with either software or dependent hardware adapters.

Setting Up iSCSI ESXi Environment

The process of preparing your iSCSI environment involves the following steps.

Step	Details
1. Set up iSCSI storage	For information, see your storage vendor documentation. In addition, follow recommendations in Best Practices for iSCSI Storage with ESXi.
2. Configure iSCSI/iSER adapters	Use an appropriate workflow to configure your adapter: ■ Set Up Independent Hardware iSCSI Adapters with ESXi ■ Configure Dependent Hardware iSCSI Adapters with ESXi ■ Configure the Software iSCSI Adapter with ESXi ■ Configure iSER with ESXi
3. Create a datastore on iSCSI storage	Creating vSphere Datastores

iSCSI Networking

For certain types of iSCSI adapters, you must configure VMkernel networking.

You can verify the network configuration by using the `vmkping` utility.

The independent hardware iSCSI adapter does not require VMkernel networking. You can configure network parameters, such as an IP address, subnet mask, and default gateway on the independent hardware iSCSI adapter.

All types of iSCSI adapters support IPv4 and IPv6 protocols.

iSCSI Adapter (vmhba)	Description	VMkernel Networking	Adapter Network Settings
Independent Hardware iSCSI Adapter	Third-party adapter that offloads the iSCSI and network processing and management from your host.	Not required.	For information, see Edit Network Settings for Hardware iSCSI.
Software iSCSI Adapter	Uses standard NICs to connect your host to a remote iSCSI target on the IP network.	Required. For information, see Setting Up Network for iSCSI and iSER with ESXi.	N/A
Dependent Hardware iSCSI Adapter	Third-party adapter that depends on VMware networking and iSCSI configuration and management interfaces.	Required For information, see Setting Up Network for iSCSI and iSER with ESXi.	N/A
VMware iSER Adapter	Uses an RDMA-capable network adapter to connect your host to a remote iSCSI target.	Required For information, see Setting Up Network for iSCSI and iSER with ESXi.	N/A

Discovery Methods

For all types of iSCSI adapters, you must set the dynamic discovery address or static discovery address. In addition, you must provide a target name of the storage system. For software iSCSI and dependent hardware iSCSI, the address is pingable using `vmkping`.

See Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host.

CHAP Authentication

Enable the CHAP parameter on the initiator and the storage system side. After authentication is enabled, it applies to all targets that are not yet discovered. It does not apply to targets that are already discovered.

See Configuring CHAP Parameters for iSCSI or iSER Storage Adapters on ESXi Host.

Read the following topics next:

- Set Up Independent Hardware iSCSI Adapters with ESXi
- Configure Dependent Hardware iSCSI Adapters with ESXi
- Configure the Software iSCSI Adapter with ESXi
- Configure iSER with ESXi
- Modify General Properties for iSCSI or iSER Adapters on ESXi Hosts
- Setting Up Network for iSCSI and iSER with ESXi
- Configure Port Binding for iSCSI or iSER on ESXi
- Using Jumbo Frames with iSCSI and iSER with ESXi
- Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host
- Configuring CHAP Parameters for iSCSI or iSER Storage Adapters on ESXi Host
- Configuring Advanced Parameters for iSCSI on ESXi Host
- Managing iSCSI Session on ESXi Host

Set Up Independent Hardware iSCSI Adapters with ESXi

An independent hardware iSCSI adapter is a specialized third-party adapter capable of accessing iSCSI storage over TCP/IP. This iSCSI adapter handles all iSCSI and network processing and management for your ESXi system.

Prerequisites

- Verify whether the adapter must be licensed.
- Install the adapter on your ESXi host.

For information about licensing, installation, and firmware updates, see vendor documentation.

The process of setting up the independent hardware iSCSI adapter includes these steps.

Step	Description
View Independent Hardware iSCSI Adapters	View an independent hardware iSCSI adapter and verify that it is correctly installed and ready for configuration.
Modify General Properties for iSCSI or iSER Adapters on ESXi Hosts	If needed, change the default iSCSI name and alias assigned to your iSCSI adapters. For the independent hardware iSCSI adapters, you can also change the default IP settings.
Edit Network Settings for Hardware iSCSI	Change default network settings so that the adapter is configured properly for the iSCSI SAN.
Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host	Set up dynamic discovery. With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.
Set Up CHAP for iSCSI or iSER Storage Adapter	If your iSCSI environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter.
Enable Jumbo Frames for Independent Hardware iSCSI	If your iSCSI environment supports Jumbo Frames, enable them for the adapter.

View Independent Hardware iSCSI Adapters

On the ESXi host, view an independent hardware iSCSI adapter and verify that it is correctly installed and ready for configuration.

After you install an independent hardware iSCSI adapter on the host, it appears on the list of storage adapters available for configuration. You can view its properties.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

   If installed, the hardware iSCSI adapter appears on the list of storage adapters.

4 Select the adapter to view.

   The default details for the adapter appear.

Adapter Information	Description
Model	Model of the adapter.
iSCSI Name	Unique name formed according to iSCSI standards that identifies the iSCSI adapter. You can edit the iSCSI name.

Adapter Information	Description
iSCSI Alias	A friendly name used instead of the iSCSI name. You can edit the iSCSI alias.
IP Address	Address assigned to the iSCSI HBA.
Targets	Number of targets accessed through the adapter.
Devices	All storage devices or LUNs the adapter can access.
Paths	All paths the adapter uses to access storage devices.

Edit Network Settings for Hardware iSCSI

After you install an independent hardware iSCSI adapter on an ESXi host, you might need to change its default network settings so that the adapter is configured properly for the iSCSI SAN.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.

4 Click the **Network Settings** tab and click **Edit**.

5 In the IPv4 settings section, deactivate IPv4 or select the method to obtain IP addresses.

Note The automatic DHCP option and static option are mutually exclusive.

Option	Description
No IPv4 settings	Deactivate IPv4.
Obtain IPv4 settings automatically	Use DHCP to obtain IP settings.
Use static IPv4 settings	Enter the IPv4 IP address, subnet mask, and default gateway for the iSCSI adapter.

6 In the IPv6 settings section, deactivate IPv6 or select an appropriate option for obtaining IPv6 addresses.

Note Automatic options and the static option are mutually exclusive.

Option	Description
No IPv6 settings	Deactivate IPv6.
Enable IPv6	Select an option for obtaining IPv6 addresses.
Obtain IPv6 addresses automatically through DHCP	Use DHCP to obtain IPv6 addresses.
Obtain IPv6 addresses automatically through Router Advertisement	Use router advertisement to obtain IPv6 addresses.

Option	Description
Override Link-local address for IPv6	Override the link-local IP address by configuring a static IP address.
Static IPv6 addresses	a Click **Add** to add a new IPv6 address. b Enter the IPv6 address and subnet prefix length, and click **OK**.

7 In the DNS settings section, provide IP addresses for a preferred DNS server and an alternate DNS server.

You must provide both values.

Configure Dependent Hardware iSCSI Adapters with ESXi

A dependent hardware iSCSI adapter in your ESXi system is a third-party adapter that depends on VMware networking, and iSCSI configuration and management interfaces provided by VMware.

An example of a dependent iSCSI adapter is a Broadcom 5709 NIC. When installed on a host, it presents its two components, a standard network adapter and an iSCSI engine, to the same port. The iSCSI engine appears on the list of storage adapters as an iSCSI adapter (vmhba).

The iSCSI adapter is enabled by default. To make it functional, you must connect it, through a virtual VMkernel adapter (vmk), to a physical network adapter (vmnic) associated with it. You can then configure the iSCSI adapter.

After you configure the dependent hardware iSCSI adapter, the discovery and authentication data is passed through the network connection. The iSCSI traffic goes through the iSCSI engine, bypassing the network.

The entire setup and configuration process for the dependent hardware iSCSI adapters involves several steps.

Step	Description
View Dependent Hardware iSCSI Adapters	View a dependent hardware iSCSI adapter to verify that it is correctly loaded.
Modify General Properties for iSCSI or iSER Adapters on ESXi Hosts	If needed, change the default iSCSI name and alias assigned to your adapter.
Determine Association Between iSCSI and Network Adapters	You must create network connections to bind dependent iSCSI and physical network adapters. To create the connections correctly, determine the name of the physical NIC with which the dependent hardware iSCSI adapter is associated.
Configure Port Binding for iSCSI or iSER on ESXi	Configure connections for the traffic between the iSCSI component and the physical network adapters. The process of configuring these connections is called port binding.
Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host	Set up dynamic discovery. With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

Step	Description
Set Up CHAP for iSCSI or iSER Storage Adapter	If your iSCSI environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter.
Set Up CHAP for Target	You can also configure different CHAP credentials for each discovery address or static target.
Enable Jumbo Frames for Networking	If your iSCSI environment supports Jumbo Frames, enable them for the adapter.

Dependent Hardware iSCSI Considerations

When you use dependent hardware iSCSI adapters with ESXi, certain considerations apply.

- When you use any dependent hardware iSCSI adapter, performance reporting for a NIC associated with the adapter might show little or no activity, even when iSCSI traffic is heavy. This behavior occurs because the iSCSI traffic bypasses the regular networking stack.
- If you use a third-party virtual switch deactivate automatic pinning. Use manual pinning instead, making sure to connect a VMkernel adapter (vmk) to an appropriate physical NIC (vmnic). For information, refer to your virtual switch vendor documentation.
- The Broadcom iSCSI adapter performs data reassembly in hardware, which has a limited buffer space. When you use the Broadcom iSCSI adapter in a congested network or under heavy load, activate flow control to avoid performance degradation.

  Flow control manages the rate of data transmission between two nodes to prevent a fast sender from overrunning a slow receiver. For best results, activate flow control at the end points of the I/O path, at the hosts and iSCSI storage systems.

  To activate flow control for the host, use the `esxcli system module parameters` command. For details, see the VMware knowledge base article at http://kb.vmware.com/kb/1013413
- Dependent hardware adapters support IPv4 and IPv6.

View Dependent Hardware iSCSI Adapters

On an ESXi host, view a dependent hardware iSCSI adapter to verify that it is correctly loaded.

If installed, the dependent hardware iSCSI adapter (vmhba#) appears on the list of storage adapters under such category as, for example, Broadcom iSCSI Adapter. If the dependent hardware adapter does not appear on the list of storage adapters, check whether it needs to be licensed. See your vendor documentation.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

4 Select the adapter (vmhba#) to view.

   The default details for the adapter appear, including the iSCSI name, iSCSI alias, and the status.

What to do next

Although the dependent iSCSI adapter is enabled by default, to make it functional, you must set up networking for the iSCSI traffic and bind the adapter to the appropriate VMkernel iSCSI port. You then configure discovery addresses and CHAP parameters.

Determine Association Between iSCSI and Network Adapters

On an ESXi host, network connections bind dependent iSCSI and physical network adapters. To create the connections correctly, you must determine the name of the physical NIC with which the dependent hardware iSCSI adapter is associated.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

4 Select the iSCSI adapter (vmhba#) and click the **Network Port Binding** tab under adapter details.

5 Click **Add**.

   The network adapter (vmnic#) that corresponds to the dependent iSCSI adapter is listed in the Physical Network Adapter column.

What to do next

If the VMkernel Adapter column is empty, create a VMkernel adapter (vmk#) for the physical network adapter (vmnic#) and then bind them to the associated dependent hardware iSCSI. See Setting Up Network for iSCSI and iSER with ESXi.

Configure the Software iSCSI Adapter with ESXi

With the software-based iSCSI implementation, you can use standard NICs to connect your host to a remote iSCSI target on the IP network. The software iSCSI adapter that is built into ESXi facilitates this connection by communicating with the physical NICs through the network stack.

When you use the software iSCSI adapters, consider the following:

- Designate a separate network adapter for iSCSI. Do not use iSCSI on 100 Mbps or slower adapters.

- Avoid hard coding the name of the software adapter, vmhbaXX, in the scripts. It is possible for the name to change from one ESXi release to another. The change might cause failures of your existing scripts if they use the hardcoded old name. The name change does not affect the behavior of the iSCSI software adapter.

The process of configuring the software iSCSI adapter involves several steps.

Step	Description
Add or Remove the Software iSCSI Adapter	Activate your software iSCSI adapter so that your host can use it to access iSCSI storage.
Modify General Properties for iSCSI or iSER Adapters on ESXi Hosts	If needed, change the default iSCSI name and alias assigned to your adapter.
Configure Port Binding for iSCSI or iSER on ESXi	Configure connections for the traffic between the iSCSI component and the physical network adapters. The process of configuring these connections is called port binding.
Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host	Set up dynamic discovery. With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.
Set Up CHAP for iSCSI or iSER Storage Adapter	If your iSCSI environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter.
Set Up CHAP for Target	You can also configure different CHAP credentials for each discovery address or static target.
Enable Jumbo Frames for Networking	If your iSCSI environment supports Jumbo Frames, enable them for the adapter.

Add or Remove the Software iSCSI Adapter

You must activate your software iSCSI adapter so that your ESXi host can use it to access iSCSI storage. If you do not need the software iSCSI adapter after activation, you can remove it.

You can activate only one software iSCSI adapter.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

If you plan to remove the software iSCSI adapter, perform these steps:

- Migrate all VMs from the iSCSI datastore.
- Unmount the datastore. See Unmount Datastores.
- Detach the iSCSI device and perform the adapter rescan. See Performing Planned Storage Device Removal.

Note If you boot from iSCSI using the software iSCSI adapter, the adapter is added and the network configuration is created at the first boot. If you remove the adapter, it is added again each time you boot the host.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Add or remove the adapter.

Option	Description
Add the software iSCSI adapter	a Under **Storage**, click **Storage Adapters**, and click **Add Software Adapter**. b From the drop-down menu, select **Software iSCSI Adapter** and confirm that you want to add the adapter. The software iSCSI adapter (vmhba#) is added and appears on the list of storage adapters. After adding the adapter, the host assigns the default iSCSI name to it. You can now complete the adapter configuration.
Remove the software iSCSI adapter	a Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to remove. b Click the **Properties** tab. c From the **Adapter Status Actions** menu, select **Disable** and confirm that you want to deactivate the adapter. The adapter will be marked for removal. The adapter will be removed from the host on next host reboot. d Reboot the host. After the reboot, the adapter no longer appears on the list of storage adapters. All virtual machines and other data on the storage devices associated with this adapter become inaccessible. You can later activate the adapter.

Configure iSER with ESXi

In addition to traditional iSCSI, ESXi supports the iSCSI Extensions for RDMA (iSER) protocol. When the iSER protocol is enabled, the iSCSI framework on the ESXi host can use the Remote Direct Memory Access (RDMA) transport instead of TCP/IP. You can configure iSER on your ESXi host.

For more information about the iSER protocol, see Using iSER Protocol with ESXi.

The entire setup and configuration process for VMware iSER involves several steps.

Step	Description
Install and View an RDMA Capable Network Adapter	To configure iSER with ESXi, you must first install an RDMA capable network adapter, for example, Mellanox Technologies MT27700 Family ConnectX-4. After you install this type of adapter, the vSphere Client displays its two components, an RDMA adapter and a physical network adapter `vmnic#`.
Enable the VMware iSER Adapter	To be able to use the RDMA capable adapter for iSCSI, use the `esxcli` to enable the VMware iSER storage component. The component appears in the vSphere Client as a `vmhba#` storage adapter under the VMware iSCSI over RDMA (iSER) Adapter category.

Step	Description
Modify General Properties for iSCSI or iSER Adapters on ESXi Hosts	If needed, change the default name and alias assigned to the iSER storage adapter `vmhba#`.
Configure Port Binding for iSCSI or iSER on ESXi	You must create network connections to bind the iSER storage adapter `vmhba#` and the RDMA capable network adapter `vmnic#`. The process of configuring these connections is called port binding. **Note** iSER does not support NIC teaming. When configuring port binding, use only one RDMA adapter per vSwitch.
Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host	Set up the dynamic or static discovery for your iSER storage adapter `vmhba#`. With the dynamic discovery, each time the initiator contacts a specified iSER storage system, it sends the SendTargets request to the system. The iSER system responds by supplying a list of available targets to the initiator. With the static discovery, you manually enter information for the targets.
Set Up CHAP for iSCSI or iSER Storage Adapter	If your environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your iSER storage adapter `vmhba#`.
Set Up CHAP for Target	You can also configure different CHAP credentials for each discovery address or static target.
Enable Jumbo Frames for Networking	If your environment supports Jumbo Frames, enable them for the iSER storage adapter `vmhba#`.

Install and View an RDMA Capable Network Adapter

ESXi supports RDMA capable network adapters, for example, Mellanox Technologies MT27700 Family ConnectX-4. After you install such adapter on your host, the vSphere Client displays its two components, an RDMA adapter and a physical network adapter.

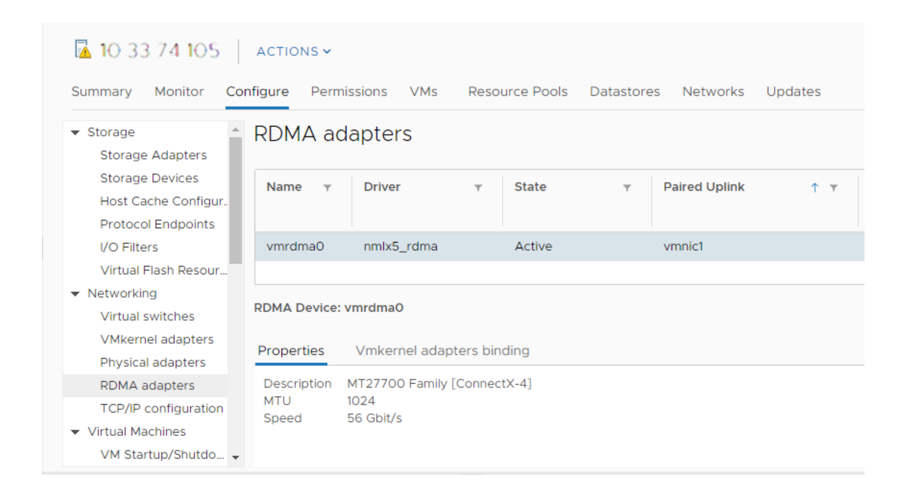
You can use the vSphere Client to view the RDMA adapter and its corresponding network adapter.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Under **Networking**, click **RDMA adapters**.

   In this example, the RDMA adapter appears on the list as `vmrdma0`. The **Paired Uplink** column displays the network component as the `vmnic1` physical network adapter.



3 To verify the description of the adapter, select the RDMA adapter from the list, and click the **Properties** tab.

Results

You can use the `vmnic#` network component of the adapter for such storage configurations as iSER or NVMe over RDMA. For the iSER configuration steps, see Configure iSER with ESXi. For information about NVMe over RDMA, see Configuring NVMe over RDMA (RoCE v2) on ESXi.

Enable the VMware iSER Adapter

To be able to use the RDMA capable adapter for iSCSI, use the `esxcli` to enable the VMware iSER storage component. After you enable the component, it appears in the vSphere Client as a `vmhba#` storage adapter under the VMware iSCSI over RDMA (iSER) Adapter category.

Prerequisites

- Make sure that your iSCSI storage supports the iSER protocol.
- Install the RDMA capable adapter on your ESXi host. For information, see Install and View an RDMA Capable Network Adapter.
- For RDMA capable adapters that support RDMA over Converged Ethernet (RoCE), determine the RoCE version that the adapter uses.
- Use the RDMA capable switch.
- Enable flow control on the ESXi host. To enable flow control for the host, use the `esxcli system module parameters` command. For details, see the VMware knowledge base article at http://kb.vmware.com/kb/1013413.
- Make sure to configure RDMA switch ports to create lossless connections between the iSER initiator and target.

Procedure

1 Use the ESXi Shell or vSphere CLI to enable the VMware iSER storage adapter and set its RoCE version.

   a Enable the iSER storage adapter.

   ```
   esxcli rdma iser add
   ```

   b Verify that the iSER adapter has been added.

   ```
   esxcli iscsi adapter list
   ```

   The output is similar to the following.

   ```
   Adapter Driver State UID Description
   ------- ------ ------- ------------- -------------------------------------
   vmhba64 iser unbound iscsi.vmhba64 VMware iSCSI over RDMA (iSER) Adapter
   ```

   c Specify the RoCE version that iSER uses to connect to the target.

   Use the RoCE version of the RDMA capable adapter. The command you enter is similar to the following:

   ```
   esxcli rdma iser params set -a vmhba64 -r 1
   ```

   When the command completes, a message similar to the following appears in the VMkernel log.

   ```
   vmkernel.0:2020-02-18T18:26:15.949Z cpu6:2100717 opID=45abe37e)iser: iser_set_roce:
   Setting roce type: 1 for vmhba: vmhba64
   vmkernel.0:2020-02-18T18:26:15.949Z cpu6:2100717 opID=45abe37e)iser: iser_set_roce:
   Setting rdma port: 3260 for vmhba: vmhba64
   ```

   If you do not specify the RoCE version, the host defaults to the highest RoCE version the RDMA capable adapter supports.

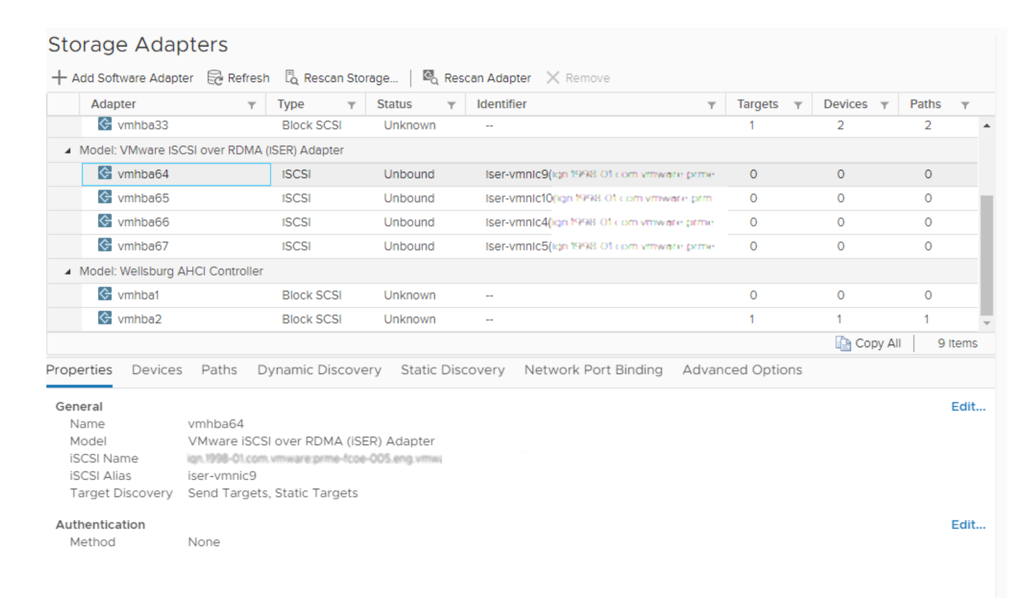
2 Use the vSphere Client to display the iSER adapter.

   a In the vSphere Client, navigate to the ESXi host.

   b Click the **Configure** tab.

   c Under **Storage**, click **Storage Adapters**, and review the list of adapters.

   If you enabled the adapter, it appears as a storage `vmhba#` adapter on the list under the VMware iSCSI over RDMA (iSER) Adapter category.



3 Select the iSER storage `vmhba#` to review its properties or perform the following tasks.

Option	Description
Configure port binding for the iSER storage adapter	You must create network connections to bind the iSER storage adapter `vmhba#` and the RDMA capable network adapter `vmnic#`. The process of configuring these connections is called port binding. For general information about port binding, see Setting Up Network for iSCSI and iSER with ESXi. To configure port binding for iSER, see Configure Port Binding for iSCSI or iSER on ESXi .
Set up dynamic or static discovery for the iSER storage adapter	For information, see Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host.
Configure the Challenge Handshake Authentication Protocol (CHAP) for the iSER storage adapter	For information, see Set Up CHAP for iSCSI or iSER Storage Adapter.

What to do next

For more information, see the VMware knowledge base article at https://kb.vmware.com/s/article/79148.

Modify General Properties for iSCSI or iSER Adapters on ESXi Hosts

You can change default name and alias assigned to your iSCSI or iSER storage adapters by the ESXi host. For the independent hardware iSCSI adapters, you can also change the default IP settings.

Important When you modify any default properties for your adapters, make sure to use correct formats for their names and IP addresses.

Prerequisites

Required privilege: **Host .Configuration.Storage Partition Configuration**

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.

4 Click the **Properties** tab, and click **Edit** in the General panel.

5 (Optional) Modify the following general properties.

Option	Description
iSCSI Name	Unique name formed according to iSCSI standards that identifies the iSCSI adapter. If you change the name, make sure that the name you enter is worldwide unique and properly formatted. Otherwise, certain storage devices might not recognize the iSCSI adapter.
iSCSI Alias	A friendly name you use instead of the iSCSI name.

Results

If you change the iSCSI name, it is used for new iSCSI sessions. For existing sessions, the new settings are not used until you log out and log in again.

What to do next

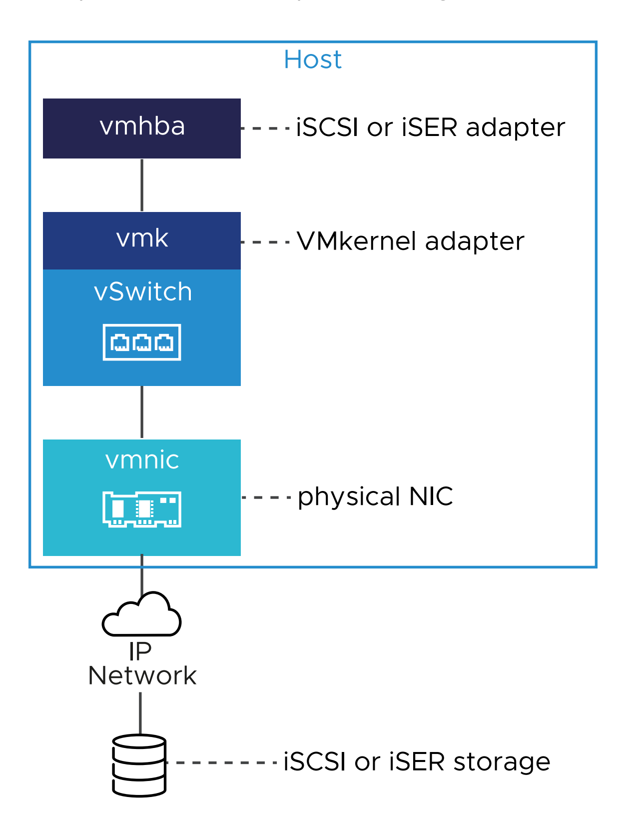
For other configuration steps you can perform for the iSCSI or iSER storage adapters, see the following topics:

- Set Up Independent Hardware iSCSI Adapters with ESXi
- Configure Dependent Hardware iSCSI Adapters with ESXi
- Configure the Software iSCSI Adapter with ESXi
- Configure iSER with ESXi

Setting Up Network for iSCSI and iSER with ESXi

Certain types of iSCSI adapters depend on the VMkernel networking. These adapters include the software or dependent hardware iSCSI adapters, and the VMware iSCSI over RDMA (iSER) adapter. If your ESXi environment includes any of these adapters, you must configure connections for the traffic between the iSCSI or iSER component and the physical network adapters.

Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You use 1:1 mapping between each virtual and physical network adapter. You then associate the VMkernel adapter with an appropriate iSCSI or iSER adapter. This process is called port binding.



Follow these rules when configuring the port binding:

- You can connect the software iSCSI adapter with any physical NICs available on your host.
- The dependent iSCSI adapters must be connected only to their own physical NICs.
- You must connect the iSER adapter only to the RDMA-capable network adapter.

For specific considerations on when and how to use network connections with software iSCSI, see the VMware knowledge base article at http://kb.vmware.com/kb/2038869.

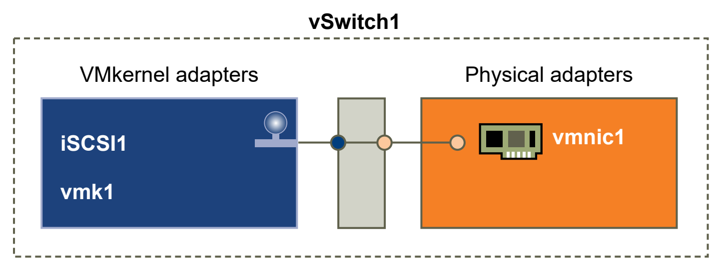
Multiple Network Adapters in iSCSI or iSER Configuration

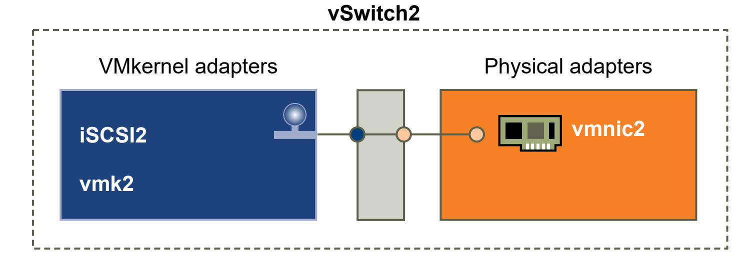
If your host has more than one physical network adapter for iSCSI or iSER, you can use the adapters for multipathing.

You can use multiple physical adapters in a single or multiple switch configurations.

In the multiple switch configuration, you designate a separate vSphere switch for each virtual-to-physical adapter pair.

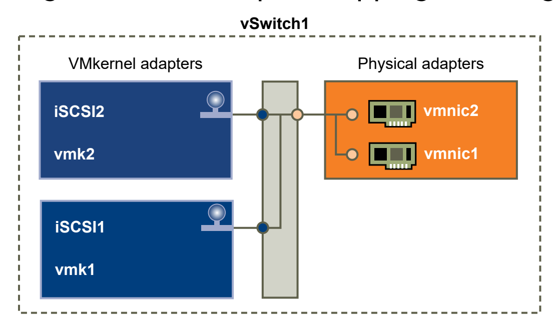
Figure 5-1. 1:1 Adapter Mapping on Separate vSphere Standard Switches





An alternative is to add all NICs and VMkernel adapters to the single vSphere switch. The number of VMkernel adapters must correspond to the number of physical adapters on the vSphere Standard switch. The single switch configuration is not appropriate for iSER because iSER does not support NIC teaming.

Figure 5-2. 1:1 Adapter Mapping on a Single vSphere Standard Switch



For that type of configuration, you must override the default network setup and make sure that each VMkernel adapter maps to only one corresponding active physical adapter, as the table indicates.

VMkernel Adapter (vmk#)	Physical Network Adapter (vmnic#)
vmk1 (iSCSI1)	**Active Adapters** vmnic1 **Unused Adapters** vmnic2
vmk2 (iSCSI2)	**Active Adapters** vmnic2 **Unused Adapters** vmnic1

You can also use distributed switches. For more information about vSphere distributed switches and how to change the default network policy, see the *vSphere Networking* documentation.

The following considerations apply when you use multiple physical adapters:

- Physical network adapters must be on the same subnet as the storage system they connect to.
- (Applies only to iSCSI and not to iSER) If you use separate vSphere switches, you must connect them to different IP subnets. Otherwise, VMkernel adapters might experience connectivity problems and the host fails to discover the LUNs.
- The single switch configuration is not appropriate for iSER because iSER does not support NIC teaming.

Do not use port binding when any of the following conditions exist:

- Array target iSCSI ports are in a different broadcast domain and IP subnet.
- VMkernel adapters used for iSCSI connectivity exist in different broadcast domains, IP subnets, or use different virtual switches.

Note In iSER configurations, the VMkernel adapters used for iSER connectivity cannot be used for converged traffic. The VMkernel adapters that you created to enable connectivity between the ESXi host with iSER and the iSER target must be used only for iSER traffic.

Best Practices for Configuring Networking with Software iSCSI

When you configure networking with software iSCSI, consider several best practices.

Software iSCSI Port Binding

You can bind the software iSCSI initiator on the ESXi host to a single or multiple VMkernel ports, so that iSCSI traffic flows only through the bound ports. Unbound ports are not used for iSCSI traffic.

When port binding is configured, the iSCSI initiator creates iSCSI sessions from all bound ports to all configured target portals.

See the following examples.

VMkernel Ports	Target Portals	iSCSI Sessions
2 bound VMkernel ports	2 target portals	4 sessions (2 x 2)
4 bound VMkernel ports	1 target portal	4 sessions (4 x 1)
2 bound VMkernel ports	4 target portals	8 sessions (2 x 4)

Note Make sure that all target portals are reachable from all VMkernel ports when port binding is used. Otherwise, iSCSI sessions might fail to create. As a result, the rescan operation might take longer than expected.

No Port Binding

If you do not use port binding, the ESXi networking layer selects the best VMkernel port based on its routing table. The host uses the port to create an iSCSI session with the target portal. Without the port binding, only one session per each target portal is created.

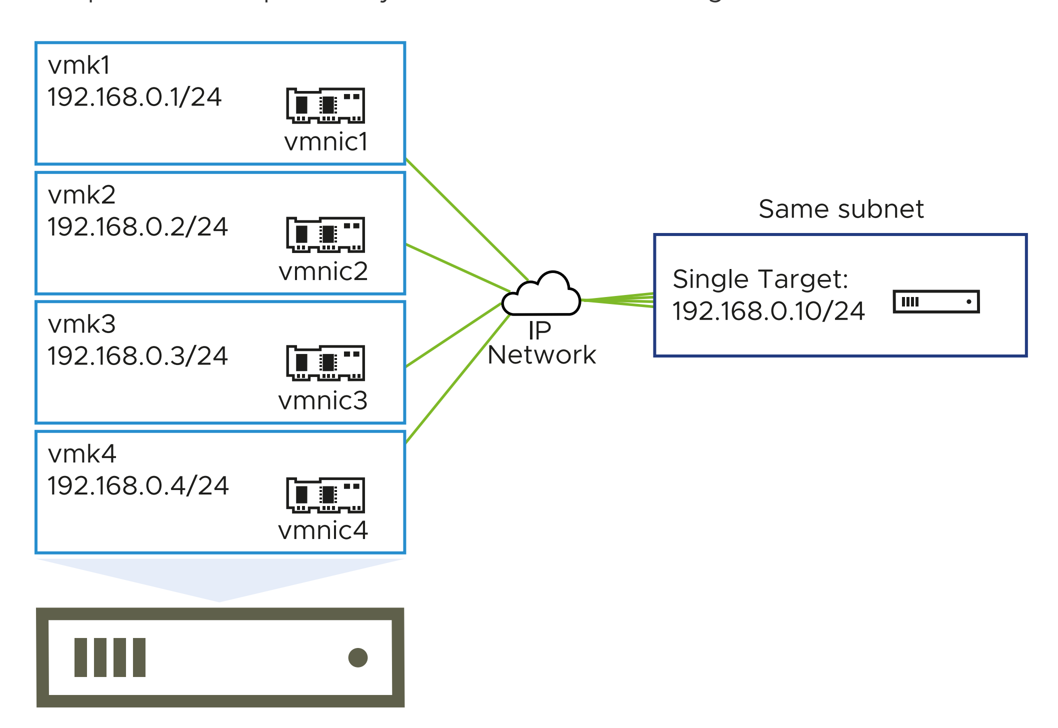
See the following examples.

VMkernel Ports	Target Portals	iSCSI Sessions
2 unbound VMkernel ports	2 target portals	2 sessions
4 unbound VMkernel ports	1 target portal	1 session
2 unbound VMkernel ports	4 target portals	4 sessions

Software iSCSI Multipathing

Example 1. Multiple paths for an iSCSI target with a single network portal

If your target has only one network portal, you can create multiple paths to the target by adding multiple VMkernel ports on your ESXi host and binding them to the iSCSI initiator.

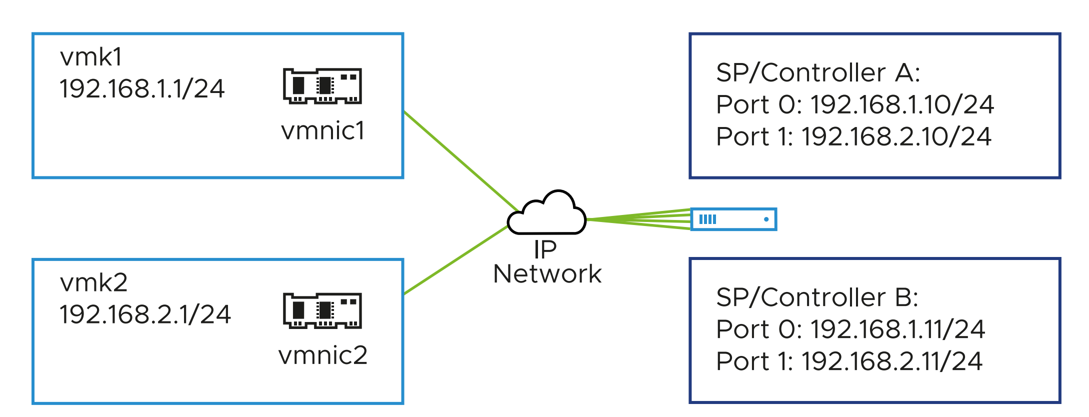


In this example, all initiator ports and the target portal are configured in the same subnet. The target is reachable through all bound ports. You have four VMkernel ports and one target portal, so total of four paths are created.

Without the port binding, only one path is created.

Example 2. Multiple paths with VMkernel ports in different subnets

You can create multiple paths by configuring multiple ports and target portals on different IP subnets. By keeping initiator and target ports in different subnets, you can force ESXi to create paths through specific ports. In this configuration, you do not use port binding because port binding requires that all initiator and target ports are on the same subnet.



ESXi selects vmk1 when connecting to Port 0 of Controller A and Controller B because all three ports are on the same subnet. Similarly, vmk2 is selected when connecting to Port 1 of Controller A and B. You can use NIC teaming in this configuration.

Total of four paths are created.

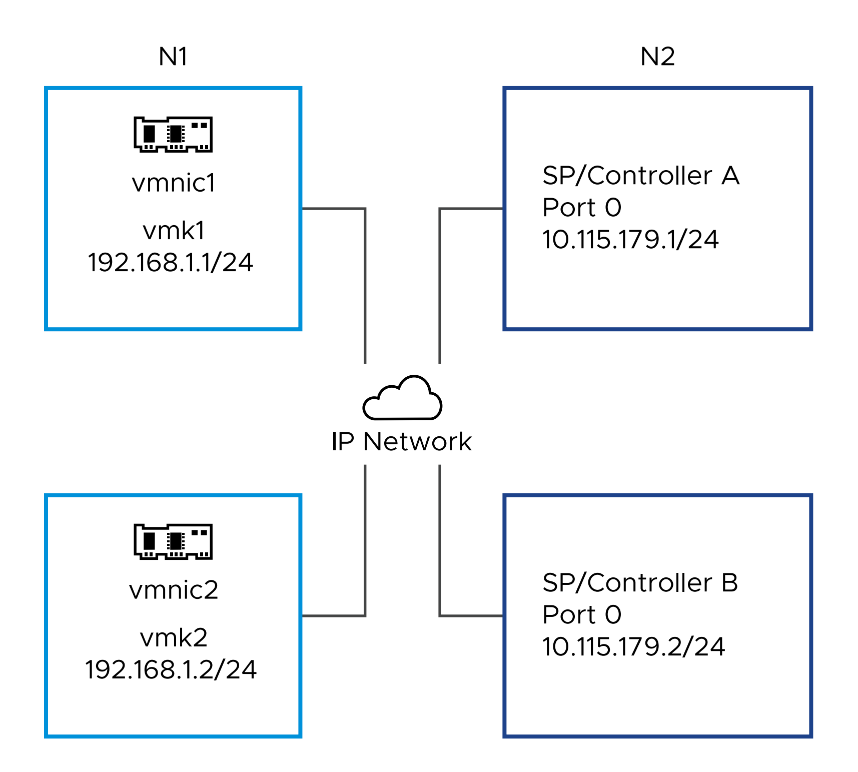
Paths	Description
Path 1	vmk1 and Port0 of Controller A
Path 2	vmk1 and Port0 of Controller B
Path 3	vmk2 and Port1 of Controller A
Path 4	vmk2 and Port1 of Controller B

Routing with Software iSCSI

You can use the `esxcli` command to add static routes for your iSCSI traffic. After you configure static routes, initiator and target ports in different subnets can communicate with each other.

Example 1. Using static routes with port binding

In this example, you keep all bound VMkernel ports in one subnet (N1) and configure all target portals in another subnet (N2). You can then add a static route for the target subnet (N2).

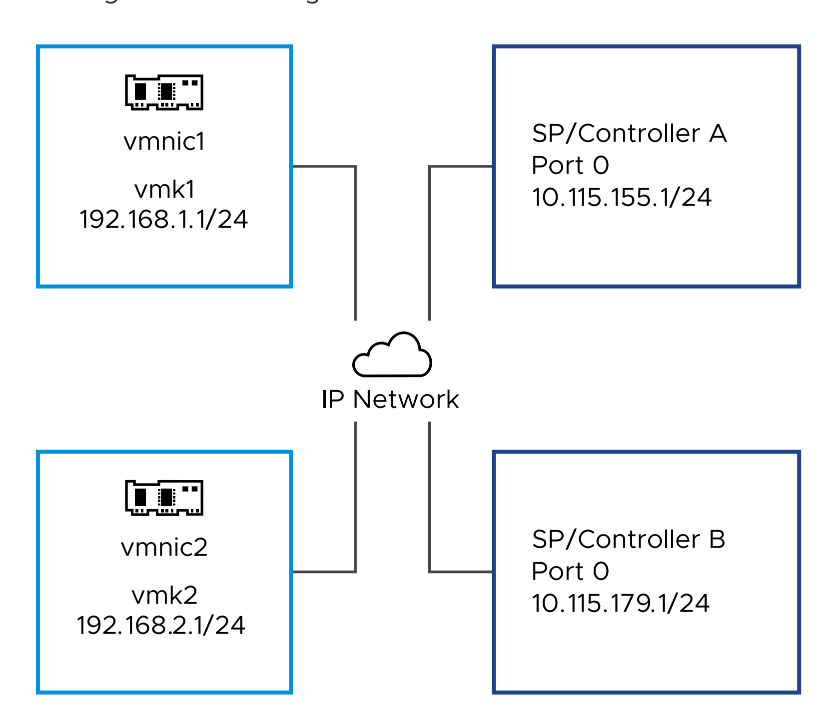


Use the following command:

```
# esxcli network ip route ipv4 add -gateway 192.168.1.253 -network
10.115.179.0/24
```

Example 2. Using static routes to create multiple paths

In this configuration, you use static routing when using different subnets. You cannot use the port binding with this configuration.



You configure vmk1 and vmk2 in separate subnets, 192.168.1.0 and 192.168.2.0. Your target portals are also in separate subnets, 10.115.155.0 and 10.155.179.0.

You can add the static route for 10.115.155.0 from vmk1. Make sure that the gateway is reachable from vmk1.

esxcli network ip route ipv4 add -gateway 192.168.1.253 -network 10.115.155.0/24

You then add static route for 10.115.179.0 from vmk2. Make sure that the gateway is reachable from vmk2.

esxcli network ip route ipv4 add -gateway 192.168.2.253 -network 10.115.179.0/24

When connecting with Port 0 of Controller A, vmk1 is used.

When connecting with Port 0 of Controller B, vmk2 is used.

Example 3. Routing with a separate gateway per vmkernel port

Starting with vSphere 6.5, you can configure a separate gateway per VMkernel port. If you use DHCP to obtain IP configuration for a VMkernel port, gateway information can also be obtained using DHCP.

To see gateway information per VMkernel port, use the following command:

esxcli network ip interface ipv4 address list

```
Name  IPv4 Address    IPv4 Netmask   IPv4 Broadcast  Address Type  Gateway         DHCP DNS
----  --------------  -------------  --------------  ------------  --------------  --------
vmk0  10.115.155.122  255.255.252.0  10.115.155.255  DHCP          10.115.155.253      true
vmk1  10.115.179.209  255.255.252.0  10.115.179.255  DHCP          10.115.179.253      true
vmk2  10.115.179.146  255.255.252.0  10.115.179.255  DHCP          10.115.179.253      true
```

With separate gateways per VMkernel port, you use port binding to reach targets in different subnets.

Managing iSCSI Network

Special considerations apply to network adapters, both physical and VMkernel, that are associated with an iSCSI adapter.

After you create network connections for iSCSI, an iSCSI indicator becomes enabled in the vSphere Client. The indicator shows that a particular virtual or physical network adapter is iSCSI-bound. To avoid disruptions in iSCSI traffic, follow these guidelines and considerations when managing iSCSI-bound virtual and physical network adapters:

- Make sure that the VMkernel network adapters are assigned addresses on the same subnet as the iSCSI storage portal they connect to.
- iSCSI adapters using VMkernel adapters cannot connect to iSCSI ports on different subnets, even if the iSCSI adapters discover those ports.

- When using separate vSphere switches to connect physical network adapters and VMkernel adapters, make sure that the vSphere switches connect to different IP subnets.
- If VMkernel adapters are on the same subnet, they must connect to a single vSwitch.
- If you migrate VMkernel adapters to a different vSphere switch, move associated physical adapters.
- Do not make configuration changes to iSCSI-bound VMkernel adapters or physical network adapters.
- Do not make changes that might break association of VMkernel adapters and physical network adapters. You can break the association if you remove one of the adapters or the vSphere switch that connects them. Or if you change the 1:1 network policy for their connection.

iSCSI Network Troubleshooting

A warning sign indicates non-compliant port group policy for an iSCSI-bound VMkernel adapter.

Problem

The VMkernel adapter's port group policy is considered non-compliant in the following cases:

- The VMkernel adapter is not connected to an active physical network adapter.
- The VMkernel adapter is connected to more than one physical network adapter.
- The VMkernel adapter is connected to one or more standby physical adapters.
- The active physical adapter is changed.

Solution

Set up the correct network policy for the iSCSI-bound VMkernel adapter. See Configure Port Binding for iSCSI or iSER on ESXi .

Configure Port Binding for iSCSI or iSER on ESXi

The port binding creates connections for the traffic between certain types of iSCSI and iSER adapters and the physical network adapters on your ESXi system.

The following types of adapters require the port binding:

- Software iSCSI adapter
- Dependent hardware iSCSI adapter
- VMware iSCSI over RDMA (iSER) adapter

The following tasks discuss the network configuration with a vSphere Standard switch and a single physical network adapter. If you have multiple network adapters, see Multiple Network Adapters in iSCSI or iSER Configuration.

Note iSER does not support NIC teaming. When configuring port binding for iSER, use only one RDMA-enabled physical adapter (vmnic#) and one VMkernel adapter (vmk#) per vSwitch.

You can also use the VMware vSphere® Distributed Switch™ and VMware NSX® Virtual Switch™ in the port binding configuration. For information about NSX virtual switches, see the *VMware NSX Data Center for vSphere* documentation.

If you use a vSphere distributed switch with multiple uplink ports, for port binding, create a separate distributed port group per each physical NIC. Then set the team policy so that each distributed port group has only one active uplink port. For detailed information on distributed switches, see the *vSphere Networking* documentation.

Create a Single VMkernel Adapter for iSCSI or iSER

Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter on your ESXi host. You then use the created VMkernel adapter in the port binding configuration with the iSCSI or iSER adapters.

Prerequisites

- If you are creating a VMkernel adapter for dependent hardware iSCSI, you must use the physical network adapter (vmnic#) that corresponds to the iSCSI component. See Determine Association Between iSCSI and Network Adapters.
- With the iSER adapter, make sure to use an appropriate RDMA-capable vmnic#. See Install and View an RDMA Capable Network Adapter.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Select **Add Networking** from the right-click menu.

3 Select **VMkernel Network Adapter**, and click **Next**.

4 Select **New standard switch** to create a vSphere Standard switch.

5 Click the **Add adapters** icon, and select an appropriate network adapter (vmnic#) to use for iSCSI.

   Make sure to assign the adapter to Active Adapters.

6 Enter a network label.

   A network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, iSCSI or iSER.

7 Specify the IP settings.

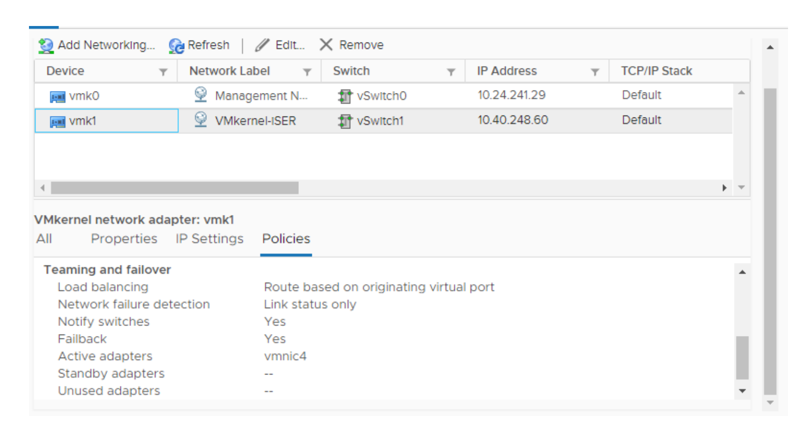
8 Review the information and click **Finish**.

You created the virtual VMkernel adapter (vmk#) for a physical network adapter (vmnic#) on your host.

9 Verify your configuration.

a Under **Networking**, select **VMkernel Adapters**, and select the VMkernel adapter (vmk#) from the list.

b Click the **Policies** tab, and verify that the corresponding physical network adapter (vmnic#) appears as an active adapter under **Teaming and failover**.



What to do next

If your host has one physical network adapter for iSCSI traffic, bind the VMkernel adapter that you created to the iSCSI or iSER vmhba adapter.

If you have multiple network adapters, you can create additional VMkernel adapters and then perform iSCSI binding. The number of virtual adapters must correspond to the number of physical adapters on the host. For information, see Multiple Network Adapters in iSCSI or iSER Configuration.

Bind iSCSI or iSER Adapters to VMkernel Adapters

On the ESXi host, bind an iSCSI or iSER adapter with a VMkernel adapter.

Prerequisites

Create a virtual VMkernel adapter for each physical network adapter on your host. If you use multiple VMkernel adapters, set up the correct network policy.

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the appropriate iSCSI or iSER adapter (vmhba#) from the list.

4 Click the **Network Port Binding** tab and click the **Add** icon.

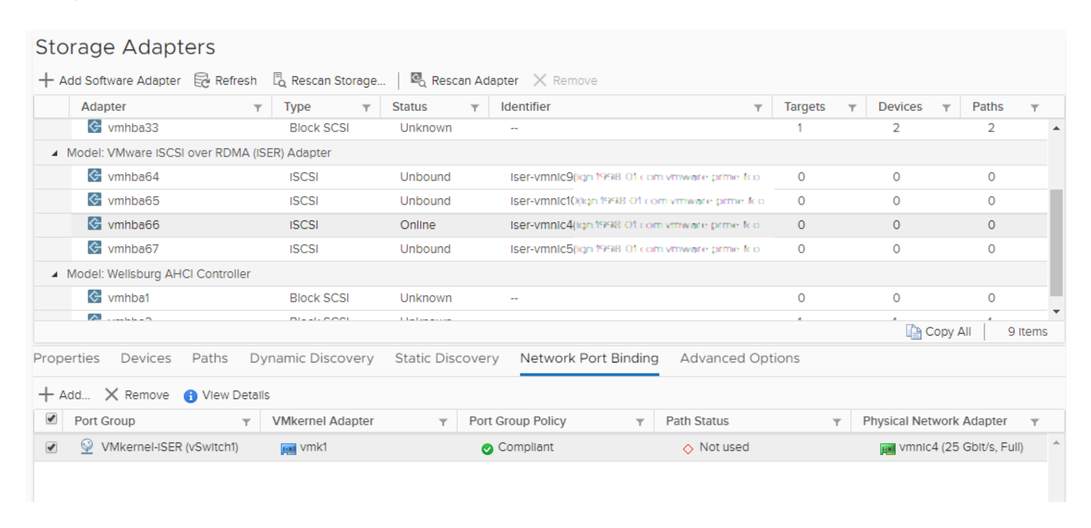
5 Select a VMkernel adapter to bind with the iSCSI or iSER adapter.

Note Make sure that the network policy for the VMkernel adapter is compliant with the binding requirements.

You can bind the software iSCSI adapter to one or more VMkernel adapters. For a dependent hardware iSCSI adapter or the iSER adapter, only one VMkernel adapter associated with the correct physical NIC is available.

6 Click **OK**.

The network connection appears on the list of network port bindings for the iSCSI or iSER adapter.



Review Port Binding Details on the ESXi Host

Review networking details of the VMkernel adapter that is bound to the iSCSI or iSER vmhba adapter.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the appropriate iSCSI or iSER adapter from the list.

4 Click the **Network Port Binding** tab and select the VMkernel adapter from the list.

5 Click the **View Details** icon.

6 Review the VMkernel adapter and physical adapter information by switching between available tabs.

What to do next

For other configuration steps you can perform for the iSCSI or iSER storage adapters, see the following topics:

- Configure Dependent Hardware iSCSI Adapters with ESXi
- Configure the Software iSCSI Adapter with ESXi
- Configure iSER with ESXi

Using Jumbo Frames with iSCSI and iSER with ESXi

ESXi supports the use of Jumbo Frames with iSCSI and iSER protocols. Jumbo Frames are Ethernet frames with the size that exceeds 1500 Bytes. The maximum transmission unit (MTU) parameter is typically used to measure the size of Jumbo Frames.

When you use Jumbo Frames for iSCSI traffic, the following considerations apply:

- All network components must support Jumbo Frames.
- Check with your vendors to ensure your physical NICs and iSCSI adapters support Jumbo Frames.
- To set up and verify physical network switches for Jumbo Frames, consult your vendor documentation.

The following table explains the level of support that ESXi provides to Jumbo Frames.

Table 5-1. Support of Jumbo Frames

Type of iSCSI Adapters	Jumbo Frames Support
Software iSCSI	Supported
Dependent Hardware iSCSI	Supported. Check with vendor.
Independent Hardware iSCSI	Supported. Check with vendor.
VMware iSER	Supported. Check with vendor.

Enable Jumbo Frames for Networking

You can enable Jumbo Frames for ESXi storage adapters that use VMkernel networking for their traffic. These adapters include software iSCSI adapters, dependent hardware iSCSI adapters, and VMware iSER adapters.

To enable Jumbo Frames, change the default value of the maximum transmission units (MTU) parameter. You change the MTU parameter on the vSphere switch that you use for iSCSI traffic. For more information, see the *vSphere Networking* documentation.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Networking**, click **Virtual switches**, and select the vSphere switch that you want to modify from the list.

4 Click the **Edit settings** icon.

5 On the Properties page, change the MTU parameter.

   This step sets the MTU for all physical NICs on that standard switch. Set the MTU value to the largest MTU size among all NICs connected to the standard switch. ESXi supports the MTU size of up to 9000 Bytes.

Enable Jumbo Frames for Independent Hardware iSCSI

To enable Jumbo Frames for independent hardware iSCSI adapters on your ESXi host, change the default value of the maximum transmission units (MTU) parameter.

Use the Advanced Options settings to change the MTU parameter for the iSCSI HBA.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the independent hardware iSCSI adapter from the list of adapters.

4 Click the **Advanced Options** tab and click **Edit**.

5 Change the value of the MTU parameter.

   ESXi supports the MTU size up to 9000 Bytes.

Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host

You need to set up target discovery addresses, so that the iSCSI or iSER storage adapter can determine which storage resource on the network is available for access on your ESXi host.

The ESXi system supports these discovery methods:

Dynamic Discovery

Also known as SendTargets discovery. Each time the initiator contacts a specified iSCSI server, the initiator sends the SendTargets request to the server. The server responds by supplying a list of available targets to the initiator. The names and IP addresses of these targets appear on the **Static Discovery** tab. If you remove a static target added by dynamic discovery, the target might be returned to the list the next time a rescan happens, the storage adapter is reset, or the host is rebooted.

Note With software and dependent hardware iSCSI, ESXi filters target addresses based on the IP family of the iSCSI server address specified. If the address is IPv4, IPv6 addresses that might come in the SendTargets response from the iSCSI server are filtered out. When DNS names are used to specify an iSCSI server, or when the SendTargets response from the iSCSI server has DNS names, ESXi relies on the IP family of the first resolved entry from DNS lookup.

Static Discovery

In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets. The iSCSI or iSER adapter uses a list of targets that you provide to contact and communicate with the iSCSI servers.

When you set up static or dynamic discovery, you can only add new iSCSI targets. You cannot change any parameters of an existing target. To make changes, remove the existing target and add a new one.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.

4 Configure the discovery method.

Discovery Method	Description
Dynamic Discovery	a Click **Dynamic Discovery** and click **Add**. b Enter the IP address or DNS name of the storage system and click **OK**. c Rescan the iSCSI adapter. After establishing the SendTargets session with the iSCSI system, your host populates the Static Discovery list with all newly discovered targets. **Note** A dynamically discovered target remains on the list even after it is removed from the array side.
Static Discovery	a Click **Static Discovery** and click **Add**. b Enter the target's information and click **OK** c Rescan the iSCSI adapter.

What to do next

For other configuration steps you can perform for the iSCSI or iSER storage adapters, see the following topics:

- Set Up Independent Hardware iSCSI Adapters with ESXi
- Configure Dependent Hardware iSCSI Adapters with ESXi
- Configure the Software iSCSI Adapter with ESXi
- Configure iSER with ESXi

Remove Dynamic or Static iSCSI Targets

Remove iSCSI servers connected to your ESXi host.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the iSCSI adapter to modify from the list.

4 Switch between **Dynamic Discovery** and **Static Discovery**.

5 Select an iSCSI server to remove and click **Remove**.

6 Rescan the iSCSI adapter.

   If you are removing the static target that was dynamically discovered, you need to remove it from the storage system before performing the rescan. Otherwise, your host will automatically discover and add the target to the list of static targets when you rescan the adapter.

Configuring CHAP Parameters for iSCSI or iSER Storage Adapters on ESXi Host

The IP networks that the iSCSI technology uses to connect your ESXi host to remote targets do not protect the data they transport. As a result, you must ensure security of the connection. One of the protocols that iSCSI implements is the Challenge Handshake Authentication Protocol (CHAP). The CHAP protocol verifies the legitimacy of ESXi initiators that access targets on the network.

CHAP uses a three-way handshake algorithm to verify the identity of your host and, if applicable, of the iSCSI target when the host and target establish a connection. The verification is based on a predefined private value, or CHAP secret, that the initiator and target share.

ESXi supports CHAP authentication at the adapter level. In this case, all targets receive the same CHAP name and secret from the iSCSI initiator. For software and dependent hardware iSCSI adapters, and for iSER adapters, ESXi also supports per-target CHAP authentication, which allows you to configure different credentials for each target to achieve greater level of security.

Selecting CHAP Authentication Method

ESXi supports unidirectional CHAP for all types of iSCSI and iSER initiators, and bidirectional CHAP for software and dependent hardware iSCSI, and for iSER.

Before configuring CHAP, check whether CHAP is activated at the iSCSI storage system. Also, obtain information about the CHAP authentication method the system supports. If CHAP is activated, configure it for your initiators, making sure that the CHAP authentication credentials match the credentials on the iSCSI storage.

ESXi supports the following CHAP authentication methods:

Unidirectional CHAP

In unidirectional CHAP authentication, the target authenticates the initiator, but the initiator does not authenticate the target.

Bidirectional CHAP

The bidirectional CHAP authentication adds an extra level of security. With this method, the initiator can also authenticate the target. VMware supports this method for software and dependent hardware iSCSI adapters, and for iSER adapters.

For software and dependent hardware iSCSI adapters, and for iSER adapters, you can set unidirectional CHAP and bidirectional CHAP for each adapter or at the target level. Independent hardware iSCSI supports CHAP only at the adapter level.

When you set the CHAP parameters, specify a security level for CHAP.

Note When you specify the CHAP security level, how the storage array responds depends on the array's CHAP implementation and is vendor-specific. For information on CHAP authentication behavior in different initiator and target configurations, consult the array documentation.

Table 5-2. CHAP Security Level

CHAP Security Level	Description	Supported Storage Adapters
None	The host does not use CHAP authentication. If authentication is activated, use this option to deactivate it.	Independent hardware iSCSI Software iSCSI Dependent hardware iSCSI iSER
Use unidirectional CHAP if required by target	The host prefers a non-CHAP connection, but can use a CHAP connection if required by the target.	Software iSCSI Dependent hardware iSCSI iSER
Use unidirectional CHAP unless prohibited by target	The host prefers CHAP, but can use non-CHAP connections if the target does not support CHAP.	Independent hardware iSCSI Software iSCSI Dependent hardware iSCSI iSER

Table 5-2. CHAP Security Level (continued)

CHAP Security Level	Description	Supported Storage Adapters
Use unidirectional CHAP	The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails.	Independent hardware iSCSI Software iSCSI Dependent hardware iSCSI iSER
Use bidirectional CHAP	The host and the target support bidirectional CHAP.	Software iSCSI Dependent hardware iSCSI iSER

Set Up CHAP for iSCSI or iSER Storage Adapter

When you set up CHAP name and secret at the iSCSI/iSER adapter level, all targets receive the same parameters from the adapter. By default, all discovery addresses or static targets inherit CHAP parameters that you set up at the adapter level.

The CHAP name cannot exceed 511 alphanumeric characters and the CHAP secret cannot exceed 255 alphanumeric characters. Some adapters, for example the QLogic adapter, might have lower limits, 255 for the CHAP name and 100 for the CHAP secret.

Prerequisites

- Before setting up CHAP parameters for software or dependent hardware iSCSI, determine whether to configure unidirectional or bidirectional CHAP. Independent hardware iSCSI adapters do not support bidirectional CHAP.
- Verify CHAP parameters configured on the storage side. Parameters that you configure must match the ones on the storage side.
- Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

1 Navigate to the iSCSI or iSER storage adapter.

   a In the vSphere Client, navigate to the ESXi host.

   b Click the **Configure** tab.

   c Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.

2 Click the **Properties** tab and click **Edit** in the **Authentication** panel.

3 Specify authentication method.

   - **None**
   - **Use unidirectional CHAP if required by target**
   - **Use unidirectional CHAP unless prohibited by target**

- **Use unidirectional CHAP**
- **Use bidirectional CHAP**. To configure bidirectional CHAP, you must select this option.

4 Specify the outgoing CHAP name.

Make sure that the name you specify matches the name configured on the storage side.

- To set the CHAP name to the iSCSI adapter name, select **Use initiator name**.
- To set the CHAP name to anything other than the iSCSI initiator name, deselect **Use initiator name** and enter a name in the **Name** text box.

5 Enter an outgoing CHAP secret to be used as part of authentication. Use the same secret that you enter on the storage side.

6 If configuring bidirectional CHAP, specify incoming CHAP credentials.

Make sure to use different secrets for the outgoing and incoming CHAP.

7 Click **OK**.

8 Rescan the iSCSI adapter.

Results

If you change the CHAP parameters, they are used for new iSCSI sessions. For existing sessions, new settings are not used until you log out and log in again.

What to do next

For other configuration steps you can perform for the iSCSI or iSER storage adapters, see the following topics:

- Set Up Independent Hardware iSCSI Adapters with ESXi
- Configure Dependent Hardware iSCSI Adapters with ESXi
- Configure the Software iSCSI Adapter with ESXi
- Configure iSER with ESXi

Set Up CHAP for Target

If you use software and dependent hardware iSCSI adapters, or an iSER storage adapter, you can configure different CHAP credentials for each discovery address or static target.

The CHAP name cannot exceed 511 and the CHAP secret 255 alphanumeric characters.

Prerequisites

- Before setting up CHAP parameters, determine whether to configure unidirectional or bidirectional CHAP.
- Verify CHAP parameters configured on the storage side. Parameters that you configure must match the ones on the storage side.
- Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

1 Navigate to the iSCSI or iSER storage adapter.

   a In the vSphere Client, navigate to the ESXi host.

   b Click the **Configure** tab.

   c Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.

2 Click either **Dynamic Discovery** or **Static Discovery**.

3 From the list of available targets, select a target to configure and click **Authentication**.

4 Deselect **Inherit settings from parent** and specify authentication method.

   - **None**
   - **Use unidirectional CHAP if required by target**
   - **Use unidirectional CHAP unless prohibited by target**
   - **Use unidirectional CHAP**
   - **Use bidirectional CHAP**. To configure bidirectional CHAP, you must select this option.

5 Specify the outgoing CHAP name.

   Make sure that the name you specify matches the name configured on the storage side.

   - To set the CHAP name to the iSCSI adapter name, select **Use initiator name**.
   - To set the CHAP name to anything other than the iSCSI initiator name, deselect **Use initiator name** and enter a name in the **Name** text box.

6 Enter an outgoing CHAP secret to be used as part of authentication. Use the same secret that you enter on the storage side.

7 If configuring bidirectional CHAP, specify incoming CHAP credentials.

   Make sure to use different secrets for the outgoing and incoming CHAP.

8 Click **OK**.

9 Rescan the storage adapter.

Results

If you change the CHAP parameters, they are used for new iSCSI sessions. For existing sessions, new settings are not used until you log out and login again.

Configuring Advanced Parameters for iSCSI on ESXi Host

You might need to configure additional parameters for iSCSI initiators on your ESXi host. For example, some iSCSI storage systems require ARP (Address Resolution Protocol) redirection to move iSCSI traffic dynamically from one port to another. In this case, you must activate the ARP redirection on your host.

The following table lists advanced iSCSI parameters that you can configure using the vSphere Client. In addition, you can use the vSphere CLI commands to configure some of the advanced parameters. For information, see the *Getting Started with ESXCLI* documentation.

Depending on the type of your adapters, certain parameters might not be available.

Important Do not change the advanced iSCSI settings unless VMware support or Storage Vendors direct you to change them.

Table 5-3. Additional Parameters for iSCSI Initiators

Advanced Parameter	Description
Header Digest	Increases data integrity. When the header digest parameter is enabled, the system performs a checksum over each header part of the iSCSI Protocol Data Unit (PDU). The system verifies the data using the CRC32C algorithm.
Data Digest	Increases data integrity. When the data digest parameter is enabled, the system performs a checksum over each PDU data part. The system verifies the data using the CRC32C algorithm. **Note** Systems that use the Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI. This offload helps to reduce the impact on performance.
ErrorRecoveryLevel	iSCSI Error Recovery Level (ERL) value that the iSCSI initiator on the host negotiates during a login.
LoginRetryMax	Maximum number of times the ESXi iSCSI initiator attempts to log into a target before ending the attempts.
MaxOutstandingR2T	Defines the R2T (Ready to Transfer) PDUs that can be in transition before an acknowledge PDU is received.
FirstBurstLength	Specifies the maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes.
MaxBurstLength	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes.
MaxRecvDataSegLength	Maximum data segment length, in bytes, that can be received in an iSCSI PDU.
MaxCommands	Maximum SCSI commands that can be queued on the iSCSI adapter.
DefaultTimeToWait	Minimum time in seconds to wait before attempting a logout or an active task reassignment after an unexpected connection termination or reset.
DefautTimeToRetain	Maximum time in seconds, during which reassigning the active task is still possible after a connection termination or reset.
LoginTimeout	Time in seconds the initiator will wait for the login response to finish.
LogoutTimeout	Time in seconds initiator will wait to get a response for Logout request PDU.
RecoveryTimeout	Specifies the amount of time, in seconds, that can lapse while a session recovery is performed. If the timeout exceeds its limit, the iSCSI initiator ends the session.
No-Op Interval	Specifies the time interval, in seconds, between NOP-Out requests sent from your iSCSI initiator to an iSCSI target. The NOP-Out requests serve as the ping mechanism to verify that a connection between the iSCSI initiator and the iSCSI target is active.

Table 5-3. Additional Parameters for iSCSI Initiators (continued)

Advanced Parameter	Description
No-Op Timeout	Specifies the amount of time, in seconds, that can lapse before your host receives a NOP-In message. The iSCSI target sends the message in response to the NOP-Out request. When the no-op timeout limit is exceeded, the initiator ends the current session and starts a new one.
ARP Redirect	With this parameter enabled, storage systems can move iSCSI traffic dynamically from one port to another. Storage systems that perform array-based failovers require the ARP parameter.
Delayed ACK	With this parameter enabled, storage systems can delay an acknowledgment of received data packets.

Configure Advanced Parameters for iSCSI on ESXi Host

The advanced iSCSI settings control such parameters as header and data digest, ARP redirection, delayed ACK, and so on.

Caution Do not make any changes to the advanced iSCSI settings unless you are working with the VMware support team or otherwise have thorough information about the values to provide for the settings.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.

4 Configure advanced parameters.

Option	Description
At the adapter level	Click the **Advanced Options** tab and click **Edit**.
At the target level	a Click either **Dynamic Discovery** or **Static Discovery**. b From the list of available targets, select a target to configure and click **Advanced**.

5 Enter any required values for the advanced parameters you want to modify.

Managing iSCSI Session on ESXi Host

To communicate with each other, iSCSI initiators on the ESXi host and targets establish iSCSI sessions. You can review and manage iSCSI sessions using vSphere CLI.

By default, software iSCSI and dependent hardware iSCSI initiators start one iSCSI session between each initiator port and each target port. If your iSCSI initiator or target has more than one port, your host can have multiple sessions established. The default number of sessions for each target equals the number of ports on the iSCSI adapter times the number of target ports.

Using vSphere CLI, you can display all current sessions to analyze and debug them. To create more paths to storage systems, you can increase the default number of sessions by duplicating existing sessions between the iSCSI adapter and target ports.

You can also establish a session to a specific target port. This capability is useful if your host connects to a single-port storage system that presents only one target port to your initiator. The system then redirects additional sessions to a different target port. Establishing a new session between your iSCSI initiator and another target port creates an additional path to the storage system.

The following considerations apply to iSCSI session management:

- Some storage systems do not support multiple sessions from the same initiator name or endpoint. Attempts to create multiple sessions to such targets can result in an unpredictable behavior of your iSCSI environment.
- Storage vendors can provide automatic session managers. Using the automatic session manages to add or delete sessions, does not guarantee lasting results and can interfere with the storage performance.

Review iSCSI Sessions

Use the vCLI command to display iSCSI sessions between an iSCSI adapter and a storage system.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

◆ To list iSCSI sessions, run the following command:

  `esxcli iscsi session list`

  The command takes these options:

Option	Description
-A|--adapter=*str*	The iSCSI adapter name, for example, vmhba34.
-s|--isid=*str*	The iSCSI session identifier.
-n|--name=*str*	The iSCSI target name, for example, iqn.X.

Add iSCSI Sessions

Use the vCLI to add an iSCSI session for a target you specify or to duplicate an existing session. By duplicating sessions, you increase the default number of sessions and create additional paths to storage systems.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To add or duplicate an iSCSI session, run the following command:

  `esxcli iscsi session add`

  The command takes these options:

Option	Description
-A|--adapter=*str*	The iSCSI adapter name, for example, vmhba34. This option is required.
-s|--isid=*str*	The ISID of a session to duplicate. You can find it by listing all sessions.
-n|--name=*str*	The iSCSI target name, for example, iqn.X.

What to do next

Rescan the iSCSI adapter.

Remove iSCSI Sessions

Use the vCLI command to remove an iSCSI session between an iSCSI adapter and a target.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To remove a session, run the following command:

  `esxcli iscsi session remove`

  The command takes these options:

Option	Description
-A|--adapter=*str*	The iSCSI adapter name, for example, vmhba34. This option is required.
-s|--isid=*str*	The ISID of a session to remove. You can find it by listing all session.
-n|--name=*str*	The iSCSI target name, for example, iqn.X.

What to do next

Rescan the iSCSI adapter.

Managing ESXi Storage Devices

6

Manage local and networked storage device that your ESXi host has access to.

For information about storage devices that ESXi supports, see What Types of Physical Storage Does ESXi Support and Target and Device Representations .

Read the following topics next:

- Device Sector Formats
- ESXi Storage Device Names and Identifiers
- Rescan Operations for ESXi Storage
- Identifying ESXi Storage Connectivity Problems
- Use the Locator LED on ESXi Storage Devices
- Erase ESXi Storage Devices
- Change Perennial Reservation Settings on ESXi Hosts

Device Sector Formats

ESXi supports storage devices with traditional and advanced sector formats. In storage, a sector is a subdivision of a track on a storage disk or device. Each sector stores a fixed amount of data.

This table introduces different storage device formats that ESXi supports.

Storage Device Format	ESXi Software Emulation	Logical Sector Size	Physical Sector Size	VMFS Datastore
512n	N/A	512	512	VMFS5 and VMFS6 (default)
512e	N/A	512	4096	VMFS5 and VMFS56 (default) **Note** Local 512e storage devices do not support VMFS5.
4Kn	512	4096	4096	VMFS6

512-Byte Native Format

ESXi supports traditional 512n storage devices that use a native 512-bytes sector size.

512-Byte Emulation Format

Due to the increasing demand for larger capacities, the storage industry has introduced advanced formats, such as 512-byte emulation, or 512e. 512e is the advanced format in which the physical sector size is 4096 bytes, but the logical sector emulates 512-bytes sector size. Storage devices that use the 512e format can support legacy applications and guest operating systems. These devices serve as an intermediate step to 4Kn sector drives.

4K Native Format with Software Emulation

Another advanced format that ESXi supports is the 4Kn sector technology. In the 4Kn devices, both physical and logical sectors are 4096 bytes (4 KiB) in length. The device does not have an emulation layer, but exposes its 4Kn physical sector size directly to ESXi.

ESXi detects and registers the 4Kn devices and automatically emulates them as 512e. The device is presented to upper layers in ESXi as 512e. But the guest operating systems always see it as a 512n device. You can continue using existing VMs with legacy guest OSes and applications on the host with the 4Kn devices.

When you use 4Kn devices, the following considerations apply:

- ESXi supports only local 4Kn SAS and SATA HDDs.
- ESXi does not support 4Kn SSD and NVMe devices, or 4Kn devices as RDMs.
- ESXi can boot only from a 4Kn device with UEFI.
- You can use the 4Kn device to configure a coredump partition and coredump file.
- Only the NMP plug-in can claim the 4Kn devices. You cannot use the HPP to claim these devices.
- With vSAN, you can use only the 4Kn capacity HDDs for vSAN Hybrid Arrays. For information, see the *Administering VMware vSAN* documentation.
- Due to the software emulation layer, the performance of the 4Kn devices depends on the alignment of the I/Os. For best performance, run workloads that issue mostly 4K aligned I/Os.
- Workloads accessing the emulated 4Kn device directly using scatter-gather I/O (SGIO) must issue I/Os compatible with the 512e disk.

Example: Determine Device Format

To determine whether the device uses the 512n, 512e, or 4Kn format, run the following command.

```
esxcli storage core device capacity list
```

The following sample output shows the format type.

```
Device                Physical Blocksize  Logical Blocksize  Logical Block Count
Size  Format Type
--------------------  ------------------  -----------------  -------------------
-----------  -----------
naa.5000xxxxxxxxx36f                 512                512           2344225968  1144641
MiB  512n
naa.5000xxxxxxxxx030                4096                512           3516328368  1716957
MiB  4Kn SWE
naa.5000xxxxxxxxx8df                 512                512           2344225968  1144641
MiB  512n
naa.5000xxxxxxxxx4f4                4096                512           3516328368  1716957
MiB  4Kn SWE
```

ESXi Storage Device Names and Identifiers

In the ESXi environment, each storage device is identified by several names.

Device Identifiers

Depending on the type of storage, the ESXi host uses different algorithms and conventions to generate an identifier for each storage device.

Storage-provided identifiers

The ESXi host queries a target storage device for the device name. From the returned metadata, the host extracts or generates a unique identifier for the device. The identifier is based on specific storage standards, is unique and persistent across all hosts, and has one of the following formats:

- naa.*xxx*
- eui.*xxx*
- t10.*xxx*

Path-based identifier

When the device does not provide an identifier, the host generates an mpx.*path* name, where *path* represents the first path to the device, for example, `mpx.vmhba1:C0:T1:L3`. This identifier can be used in the same way as the storage-provided identifies.

The mpx.*path* identifier is created for local devices on the assumption that their path names are unique. However, this identifier is not unique or persistent, and can change after every system restart.

Typically, the path to the device has the following format:

vmhba*Adapter*:C*Channel*:T*Target*:L*LUN*

- vmhba*Adapter* is the name of the storage adapter. The name refers to the physical adapter on the host, not to the SCSI controller used by the virtual machines.
- C*Channel* is the storage channel number.

  Software iSCSI adapters and dependent hardware adapters use the channel number to show multiple paths to the same target.
- T*Target* is the target number. Target numbering is determined by the host and might change when the mappings of targets visible to the host change. Targets that are shared by different hosts might not have the same target number.
- L*LUN* is the LUN number that shows the position of the LUN within the target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).

For example, `vmhba1:C0:T3:L1` represents LUN1 on target 3 accessed through the storage adapter vmhba1 and channel 0.

Legacy identifier

In addition to the device-provided identifiers or mpx.*path* identifiers, ESXi generates an alternative legacy name for each device. The identifier has the following format:

vml.*number*

The legacy identifier includes a series of digits that are unique to the device. The identifier can be derived in part from the metadata obtained through the SCSI INQUIRY command. For nonlocal devices that do not provide SCSI INQUIRY identifiers, the vml.*number* identifier is used as the only available unique identifier.

Example: Displaying Device Names in the vSphere CLI

You can use the `esxcli storage core device list` command to display all device names in the vSphere CLI. The output is similar to the following example:

```
# esxcli storage core device list
naa.XXX
      Display Name: DGC Fibre Channel Disk(naa.XXX)
      ...
      Other UIDs: vml.000XXX
mpx.vmhba1:C0:T0:L0
   Display Name: Local VMware Disk (mpx.vmhba1:C0:T0:L0)
   ...
   Other UIDs: vml.0000000000XYZ
```

NVMe Devices with NGUID Device Identifiers

For NVMe devices, ESXi generates device identifiers based on the information it retrieves from the devices. Generally, the NVMe devices support identifiers in EUI64 or NGUID formats, or use

both formats. NGUID is a Namespace Globally Unique Identifier that uses the EUI64 16-byte designator format.

For the devices that support only NGUID format, the host creates two identifiers: eui.*xxx (NGUID)* as primary and t10.*xxx_controller_serial_number* as alternative primary.

ID Formats Supported by Device		Device Identifier Generated by Host
EUI64 ID Format	**NGUID ID Format**	**ESXi 8.0**
yes	yes	t10.*xxx_EUI64*
yes	no	t10.*xxx_EUI64*
no	yes	eui.*xxx (NGUID)* as primary ID t10.*xxx_controller_serial_number* as alternative primary ID

Verify Mapping Between Primary and Alternative Device Identifiers

Use the `esxcli storage core device uidmap list` command to verify the device identifiers. The output is similar to the following:

```
 esxcli storage core device uidmap list
eui.0000xyz.....
   Primary UID: eui.0000xyz.....
   Alternative Primary UIDs: t10.0000abc....
   Legacy UID: vml.0000000000766d68....
   Alternative Legacy UIDs: vml.000000000080906....
```

Rename Storage Devices

The ESXi host assigns a display name to a storage device based on the storage type and manufacturer. You can change the display name of the device.

You cannot rename certain types of local devices.

Procedure

1 In the vSphere Client, navigate to the ESXi host.
2 Click the **Configure** tab.
3 Under **Storage**, click **Storage Devices**.
4 Select the device to rename and click **Rename**.
5 Change the device name to a friendly name.

Rescan Operations for ESXi Storage

When you perform storage management tasks or make changes in the SAN configuration, you might need to rescan storage available to your ESXi host.

When you perform VMFS datastore management operations, such as creating a VMFS datastore or RDM, adding an extent, and increasing or deleting a VMFS datastore, your host or the vCenter Server automatically rescans and updates your storage. You can deactivate the automatic rescan feature by turning off the Host Rescan Filter. See Manage Filters for LUNs and vSphere VMFS Datastores.

In certain cases, you need to perform a manual rescan. You can rescan all storage available to your host or to all hosts in a folder, cluster, and data center.

If the changes you make are isolated to storage connected through a specific adapter, perform a rescan for this adapter.

Perform the manual rescan each time you make one of the following changes.

- Zone a new disk array on a SAN.
- Create new LUNs on a SAN.
- Change the path masking on a host.
- Reconnect a cable.
- Change CHAP settings (iSCSI only).
- Add or remove discovery or static addresses (iSCSI only).
- Add a single host to the vCenter Server after you have edited or removed from the vCenter Server a datastore shared by the vCenter Server hosts and the single host.

Important If you rescan when a path is unavailable, the host removes the path from the list of paths to the device. The path reappears on the list as soon as it becomes available and starts working again.

Perform Storage Rescan

When you make changes in your SAN configuration, you might need to rescan storage. You can rescan all storage available to your ESXi host, cluster, or data center. If the changes you make are isolated to storage accessed through a specific host, perform the rescan for only this host.

Procedure

1 In the vSphere Client object navigator, browse to a host, a cluster, a data center, or a folder that contains hosts.

2 From the right-click menu, select **Storage > Rescan Storage**.

3 Specify extent of rescan.

Option	Description
Scan for New Storage Devices	Rescan all adapters to discover new storage devices. If new devices are discovered, they appear in the device list.
Scan for New VMFS Volumes	Rescan all storage devices to discover new datastores that have been added since the last scan. Any new datastores appear in the datastore list.

Perform Adapter Rescan

When you make changes in your SAN configuration and these changes are isolated to storage accessed through a specific adapter on ESXi host, perform rescan for only this adapter.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the adapter to rescan from the list.

4 Click the **Rescan Adapter** icon.

Change the Number of Scanned Storage Devices

The range of scanned LUN IDs for an ESXi host can be from 0 to 16,383. ESXi ignores LUN IDs greater than 16,383. The configurable `Disk.MaxLUN` parameter controls the range of scanned LUN ID range. The parameter has a default value of 1024.

The `Disk.MaxLUN` parameter also determines how many LUNs the SCSI scan code attempts to discover using individual INQUIRY commands if the SCSI target does not support direct discovery using REPORT_LUNS.

You can modify the `Disk.MaxLUN` parameter depending on your needs. For example, if your environment has a smaller number of storage devices with LUN IDs from 1 through 100, set the value to 101. As a result, you can improve device discovery speed on targets that do not support REPORT_LUNS. Lowering the value can shorten the rescan time and boot time. However, the time to rescan storage devices might also depend on other factors, including the type of the storage system and the load on the storage system.

In other cases, you might need to increase the value if your environment uses LUN IDs that are greater than 1023.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 In the Advanced System Settings table, select **Disk.MaxLUN** and click the **Edit** icon.

5 Change the existing value to the value of your choice, and click **OK**.

The value you enter specifies the LUN ID that is after the last one you want to discover.

For example, to discover LUN IDs from 1 through 100, set **Disk.MaxLUN** to 101.

Identifying ESXi Storage Connectivity Problems

When your ESXi connects to a storage device, it might experiences a connectivity problem. Storage connectivity problems might be caused by various reasons. Although ESXi cannot always detect the reason for a device or its paths being unavailable, the host can determine whether the problem is permanent or temporary. In other words, the host can differentiate between a permanent device loss (PDL) state of the device and a transient all paths down (APD) state of storage.

Permanent Device Loss (PDL)

A condition that occurs when a storage device permanently fails or is administratively removed or excluded. It is not expected to become available. When the device becomes permanently unavailable, ESXi receives appropriate sense codes or a login rejection from storage arrays, and is able to recognize that the device is permanently lost.

All Paths Down (APD)

A condition that occurs when a storage device becomes inaccessible to the host and no paths to the device are available. ESXi treats this as a transient condition because typically the problems with the device are temporary and the device is expected to become available again.

Connectivity Problems and vSphere High Availability

When the device enters the PDL or APD state, vSphere High Availability (HA) can detect connectivity problems and provide automated recovery for affected virtual machines on the ESXi host. vSphere HA uses VM Component Protection (VMCP) to protect virtual machines running on the host in the vSphere HA cluster against accessibility failures. For more information about VMCP and how to configure responses for datastores and virtual machines when the APD or PDL condition occurs, see the *vSphere Availability* documentation.

Detecting PDL Conditions

A storage device is considered to be in the permanent device loss (PDL) state when it becomes permanently unavailable to your ESXi host.

Typically, the PDL condition occurs when a device is unintentionally removed, or its unique ID changes, or when the device experiences an unrecoverable hardware error.

When the storage array determines that the device is permanently unavailable, it sends SCSI error sense codes or NVMe error codes to the ESXi host. After receiving these errors, your host recognizes the device as failed and registers the state of the device as PDL. For the device to be considered permanently lost, the sense codes must be received on all its paths.

After registering the PDL state of the device, the host stops attempts to reestablish connectivity or to send commands to the device.

The vSphere Client displays the following information for the device:

- The operational state of the device changes to `Lost Communication`.
- All paths are shown as `Dead`.
- Datastores on the device are not available.

If no open connections to the device exist, or after the last connection closes, the host removes the PDL device and all paths to the device. You can deactivate the automatic removal of paths by setting the advanced host parameter `Disk.AutoremoveOnPDL` to 0.

If the device returns from the PDL condition, the host can discover it, but treats it as a new device. Data consistency for virtual machines on the recovered device is not guaranteed.

Note When a device fails without sending appropriate SCSI sense codes or NVMe error codes or an iSCSI login rejection, the host cannot detect PDL conditions. In this case, the host continues to treat the device connectivity problems as APD even when the device fails permanently.

Permanent Device Loss and SCSI Sense Codes

The following VMkernel log example of a SCSI sense code indicates that the device is in the PDL state.

```
H:0x0 D:0x2 P:0x0 Valid sense data: 0x5 0x25 0x0 or Logical Unit Not Supported
```

Permanent Device Loss and NVMe Error Codes

The following VMkernel log example of an NVMe error code indicates that the device is in the PDL state.

```
H:0x0 D:0xb P:0x0 or H:0x0 D:0x11a P:0x0
```

Permanent Device Loss and iSCSI

On iSCSI arrays with a single LUN per target, PDL is detected through iSCSI login failure. An iSCSI storage array rejects your host's attempts to start an iSCSI session with a reason `Target Unavailable`. As with the sense codes, this response must be received on all paths for the device to be considered permanently lost.

Permanent Device Loss and Virtual Machines

After registering the PDL state of the device, the host closes all I/O from virtual machines. vSphere HA can detect PDL and restart failed virtual machines.

Performing Planned Storage Device Removal

When a storage device is malfunctioning, you can avoid permanent device loss (PDL) conditions or all paths down (APD) conditions. Perform a planned removal and reconnection of a storage device.

Planned device removal is an intentional disconnection of a storage device. You might also plan to remove a device for such reasons as upgrading your hardware or reconfiguring your storage devices. When you perform an orderly removal and reconnection of a storage device, you complete a number of tasks.

Task	Description
Migrate virtual machines from the device you plan to detach.	*vCenter Server and Host Management*
Unmount the datastore deployed on the device.	See Unmount Datastores.
Detach the storage device.	See Detach Storage Devices.
For an iSCSI device with a single LUN per target, delete the static target entry from each iSCSI HBA that has a path to the storage device.	See Remove Dynamic or Static iSCSI Targets.
Perform any necessary reconfiguration of the storage device by using the array console.	See your vendor documentation.
Reattach the storage device.	See Attach Storage Devices.
Mount the datastore and restart the virtual machines.	See Mount Datastores.

Detach Storage Devices

Safely detach a storage device from your ESXi host.

You might need to detach the device to make it inaccessible to your host, when, for example, you perform a hardware upgrade on the storage side.

Prerequisites

- The device does not contain any datastores.
- No virtual machines use the device as an RDM disk.
- The device does not contain a diagnostic partition or a scratch partition.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 Select the device to detach and click the **Detach** icon.

Results

The device becomes inaccessible. The operational state of the device changes to Unmounted.

What to do next

If multiple hosts share the device, detach the device from each host.

Attach Storage Devices

Reattach a storage device that you previously detached from the ESXi host.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 Select the detached storage device and click the **Attach** icon.

Results

The device becomes accessible.

Recovering from PDL Conditions

An unplanned permanent device loss (PDL) condition occurs when a storage device becomes permanently unavailable without being properly detached from the ESXi host.

The following items in the vSphere Client indicate that the device is in the PDL state:

- The datastore deployed on the device is unavailable.
- Operational state of the device changes to Lost Communication.
- All paths are shown as Dead.
- A warning about the device being permanently inaccessible appears in the VMkernel log file.

To recover from the unplanned PDL condition and remove the unavailable device from the host, perform the following tasks.

Task	Description
Power off and unregister all virtual machines that are running on the datastore affected by the PDL condition.	See *vSphere Virtual Machine Administration*.
Unmount the datastore.	See Unmount Datastores.
Rescan all ESXi hosts that had access to the device. **Note** If the rescan is not successful and the host continues to list the device, some pending I/O or active references to the device might still exist. Check for any items that might still have an active reference to the device or datastore. The items include virtual machines, templates, ISO images, raw device mappings, and so on.	See Perform Storage Rescan.

Handling Transient APD Conditions

A storage device is considered to be in the all paths down (APD) state when it becomes unavailable to your ESXi host for an unspecified time period.

The reasons for an APD state can be, for example, a failed switch or a disconnected storage cable.

In contrast with the permanent device loss (PDL) state, the host treats the APD state as transient and expects the device to be available again.

The host continues to retry issued commands in an attempt to reestablish connectivity with the device. If the host's commands fail the retries for a prolonged period, the host might be at risk of having performance problems. Potentially, the host and its virtual machines might become unresponsive.

To avoid these problems, your host uses a default APD handling feature. When a device enters the APD state, the host turns on a timer. With the timer on, the host continues to retry non-virtual machine commands for a limited time period only.

By default, the APD timeout is set to 140 seconds. This value is typically longer than most devices require to recover from a connection loss. If the device becomes available within this time, the host and its virtual machine continue to run without experiencing any problems.

If the device does not recover and the timeout ends, the host stops its attempts at retries and stops any non-virtual machine I/O. Virtual machine I/O continues retrying. The vSphere Client displays the following information for the device with the expired APD timeout:

- The operational state of the device changes to `Dead or Error`.
- All paths are shown as `Dead`.
- Datastores on the device are dimmed.

Even though the device and datastores are unavailable, virtual machines remain responsive. You can power off the virtual machines or migrate them to a different datastore or host.

If later the device paths become operational, the host can resume I/O to the device and end the special APD treatment.

Deactivate Storage APD Handling

The storage all paths down (APD) handling on your ESXi host is activated by default. When this functionality is activated and a storage device enters the APD state, the host continues to retry nonvirtual machine I/O commands only for a limited time period. After the time period expires, the host stops its retry attempts and terminates any nonvirtual machine I/O. You can deactivate the APD handling feature on your host.

If you deactivate the APD handling, the host will indefinitely continue to retry issued commands in an attempt to reconnect to the APD device. This behavior might cause virtual machines on the host to exceed their internal I/O timeout and become unresponsive or fail. The host might become disconnected from vCenter Server.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 In the Advanced System Settings table, select the **Misc.APDHandlingEnable** parameter and click the `Edit` icon.

5 Change the value to 0.

Results

If you deactivate the APD handling, you can reactivate it and set its value to 1 when a device enters the APD state. The internal APD handling feature turns on immediately and the timer starts with the current timeout value for each device in APD.

Change Timeout Limits for Storage APD

The timeout parameter controls how many seconds the ESXi host must retry the I/O commands to a storage device in an all paths down (APD) state. You can change the default timeout value.

The timeout period begins immediately after the device enters the APD state. After the timeout ends, the host marks the APD device as unreachable. The host stops its attempts to retry any I/O that is not coming from virtual machines. The host continues to retry virtual machine I/O.

By default, the timeout parameter on your host is set to 140 seconds. You can increase the value of the timeout if, for example, storage devices connected to your ESXi host take longer than 140 seconds to recover from a connection loss.

Note If you change the timeout parameter after the device becomes unavailable, the change does not take effect for that particular APD incident.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 In the Advanced System Settings table, select the **Misc.APDTimeout** parameter and click the `Edit` icon.

5 Change the default value.

   You can enter a value between 20 and 99999 seconds.

Verify the Connection Status of a Storage Device on ESXi Host

Use the `esxcli` command to verify the connection status of a particular storage device.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Run the **esxcli storage core device list -d=*device_ID*** command.

2 Review the connection status in the `Status:` area.

- `on` - Device is connected.
- `dead` - Device has entered the APD state. The APD timer starts.
- `dead timeout` - The APD timeout has expired.
- `not connected` - Device is in the PDL state.

Use the Locator LED on ESXi Storage Devices

Use the locator LED to identify specific storage devices connected to your ESXi host, so that you can locate them among other devices. You can turn the locator LED on or off.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 From the list of storage devices, select one or more disks and set the locator LED indicator.

- Click the **Turn On LED** icon.
- Click the **Turn Off LED** icon.

Erase ESXi Storage Devices

Certain functionalities, such as vSAN or virtual flash resource, require that your ESXi host uses clean storage devices. You can erase an HHD or flash device and remove all existing data.

Prerequisites

- Make sure that the host is in the connected state.
- Verify that the devices you plan to erase are not in use.
- Required privilege: **Host.Config.Storage**

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 Select one or more devices and click the **Erase Partitions** icon.

5 Verify that the partition information you are erasing is not critical.

6 Click **OK** to confirm your change.

Change Perennial Reservation Settings on ESXi Hosts

You can adjust the perennial reservation setting on ESXi storage devices that are used as physical Raw Device Mappings (RDMs) in Windows Server Failover Clustering (WSFC) configurations.

WSFC cluster nodes that are spread over several ESXi hosts require physical RDMs. The RDMs are shared among all hosts where cluster nodes run. The host with the active node holds persistent SCSI-3 reservations on all shared RDM devices. When the active node is running and devices are locked, no other host can write to the devices. If another participating host boots while the active node is holding the lock on the devices, the boot might take unusually long time because the host unsuccessfully attempts to contact the locked devices. The same issue might also affect rescan operations.

To prevent this problem, activate perennial reservation for all devices on the ESXi hosts where secondary WSFC nodes with RDMs reside. This setting informs the ESXi host about the permanent SCSI reservation on the devices, so that the host can skip the devices during the boot or storage rescan process.

If you later re-purpose the marked devices as VMFS datastores, remove the reservation to avoid unpredictable datastore behavior.

For information about WSFC clusters, see the *Setup for Windows Server Failover Clustering* documentation.

Prerequisites

Before marking a device as perennially reserved, make sure the device does not contain a VMFS datastore.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 From the list of storage devices, select the device and click one of the following icons.

Option	Description
Mark as Perennially Reserved	Mark the selected device as perennially reserved. **Note** Repeat the procedure for each RDM device that is participating in the WSFC cluster.
Unmark as Perennially Reserved	Clear perennial reservation for the device that was previously marked.

Results

The configuration is permanently stored with the ESXi host and persists across restarts.

Example

You can also use the `esxcli` command to mark the devices participating in the WSFC cluster.

1 Mark the devices as perennially reserved.

```
esxcli storage core device setconfig -d naa.id --perennially-reserved=true
```

2 Verify that the device is perennially reserved.

```
esxcli storage core device list -d naa.id
```

In the output of the `esxcli` command, search for the entry `Is Perennially Reserved: true`.

3 To remove the perennially reserved flag, run the following command.

```
esxcli storage core device setconfig -d naa.id --perennially-reserved=false
```

Using Flash Devices with ESXi Hosts

7

In addition to regular storage hard disk drives (HDDs), ESXi supports flash storage devices.

Unlike regular HDDs that are electromechanical devices containing moving parts, flash devices use semiconductors as their storage medium and have no moving parts. Typically, the flash devices are resilient and provide faster access to data.

To detect flash devices, ESXi uses an inquiry mechanism based on T10 standards. Check with your vendor whether your storage array supports the ESXi mechanism of flash device detection.

After the host detects the flash devices, you can use them for several tasks and functionalities.

If you use NVMe storage, enable the high-performance plug-in (HPP) to improve your storage performance. See VMware High Performance Plug-In and Path Selection Schemes.

For specifics about using NVMe storage with ESXi, see Chapter 8 About VMware NVMe Storage.

Table 7-1. Using Flash Devices with ESXi

Functionality	Description
vSAN	vSAN requires flash devices. For more information, see the *Administering VMware vSAN* documentation.
VMFS Datastores	Create VMFS datastores on flash devices. Use the datastores for the following purposes: ■ Store virtual machines. Certain guest operating systems can identify virtual disks stored on these datastores as flash virtual disks. ■ Allocate datastore space for the ESXi host swap cache. See Configure Host Cache with VMFS Datastore in vSphere Environment.
Virtual Flash Resource (VFFS)	If required by your vendor, set up a virtual flash resource and use it for I/O caching filters. See Chapter 15 Filtering Virtual Machine I/O in vSphere.

Flash Devices and Virtual Machines

Guest operating systems can identify virtual disks that reside on flash-based datastores as flash virtual disks.

Guest operating systems can use standard inquiry commands such as SCSI VPD Page (B1h) for SCSI devices and ATA IDENTIFY DEVICE (Word 217) for IDE devices.

For linked clones, native snapshots, and delta-disks, the inquiry commands report the virtual flash status of the base disk.

Operating systems can detect that a virtual disk is a flash disk under the following conditions:

- Detection of flash virtual disks is supported on VMs with virtual hardware version 8 or later.
- Devices backing a shared VMFS datastore must be marked as flash on all hosts.
- If the VMFS datastore includes several device extents, all underlying physical extents must be flash-based.

Read the following topics next:

- Best Practices for Flash Devices on ESXi Hosts
- Marking Storage Devices on ESXi Hosts
- About Virtual Flash Resource in vSphere Environment
- Configure Host Cache with VMFS Datastore in vSphere Environment

Best Practices for Flash Devices on ESXi Hosts

Follow these best practices when you use flash devices in vSphere environment.

- Use flash devices approved by the *VMware Compatibility Guide*.
- Make sure to use the latest firmware with flash devices. Frequently check with your storage vendors for any updates.
- Carefully monitor how intensively you use the flash device and calculate its estimated lifetime. The lifetime expectancy depends on how actively you continue to use the flash device. See Estimate Lifetime of Flash Devices.
- If you use NVMe devices for storage, enable the high-performance plug-in (HPP) to improve your storage performance. For specifics of using the NVMe devices, see VMware High Performance Plug-In and Path Selection Schemes

Monitor Flash Devices

You can monitor certain critical flash device parameters, including `Media Wearout Indicator`, `Temperature`, and `Reallocated Sector Count`, from an ESXi host.

Use the `esxcli` command to monitor flash devices.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- Display the flash device statistics by running the following command:

  esxcli storage core device smart get -d=*flash device_ID*

Estimate Lifetime of Flash Devices

When working with flash devices, monitor how actively you use them and calculate their estimated lifetime.

Typically, storage vendors provide reliable lifetime estimates for a flash device under ideal conditions. For example, a vendor might guarantee a lifetime of 5 years under the condition of 20 GB writes per day. However, the more realistic life expectancy of the device depends on how many writes per day your ESXi host actually generates. Follow these steps to calculate the lifetime of the flash device.

Prerequisites

Note the number of days passed since the last reboot of your ESXi host. For example, ten days.

Procedure

1 Obtain the total number of blocks written to the flash device since the last reboot.

  Run the **esxcli storage core device stats get -d=*device_ID*** command. For example:

```
~ # esxcli storage core device stats get -d t10.xxxxxxxxxxxxxxx
    Device: t10.xxxxxxxxxxxxxxx
    Successful Commands: xxxxxxx
    Blocks Read: xxxxxxxx
    Blocks Written: 629145600
    Read Operations: xxxxxxxx
```

  The Blocks Written item in the output shows the number of blocks written to the device since the last reboot. In this example, the value is 629,145,600. After each reboot, it resets to 0.

2 Calculate the total number of writes and convert to GB.

  One block is 512 bytes. To calculate the total number of writes, multiply the Blocks Written value by 512, and convert the resulting value to GB.

  In this example, the total number of writes since the last reboot is approximately 322 GB.

3 Estimate the average number of writes per day in GB.

  Divide the total number of writes by the number of days since the last reboot.

  If the last reboot was ten days ago, you get 32 GB of writes per day. You can average this number over the time period.

4 Estimate lifetime of your device by using the following formula:

vendor provided number of writes per day times *vendor provided life span* divided by *actual average number of writes per day*

For example, if your vendor guarantees a lifetime of 5 years under the condition of 20 GB writes per day, and the actual number of writes per day is 30 GB, the life span of your flash device will be approximately 3.3 years.

Keeping Flash Disks VMFS-Free

If you use the auto-partitioning boot option when installing or auto-deploying ESXi, the auto-partitioning option creates a VMFS datastore on your host's local storage. In certain cases, you need to keep your local storage flash disks unformatted.

Problem

By default, auto-partitioning deploys VMFS file systems on any unused local storage disks on your host, including flash disks.

However, a flash disk formatted with VMFS becomes unavailable for such features as virtual flash and vSAN. Both features require an unformatted flash disk and neither can share the disk with any other file system.

Solution

To ensure that auto-partitioning does not format the flash disk with VMFS, use the following boot options when you install ESXi or boot the ESXi host for the first time:

- **`autoPartition=TRUE`**
- **`skipPartitioningSsds=TRUE`**

If you use Auto Deploy, set these parameters on a reference ESXi host.

1 In the vSphere Client, navigate to the host to use as a reference host and click the **Configure** tab.

2 Click **System** to open the system options, and click **Advanced System Settings**.

3 Set the following items.

Parameter	Value
VMkernel.Boot.autoPartition	True
VMkernel.Boot.skipPartitioningSsds	True

4 Reboot the host.

If flash disks that you plan to use with the virtual flash resource and vSAN already have VMFS datastores, remove the datastores.

Marking Storage Devices on ESXi Hosts

When you configure vSAN or set up a virtual flash resource, your storage environment must include local flash devices. However, ESXi might not recognize certain storage devices as flash devices when their vendors do not support automatic flash device detection. In other cases, certain devices might not be detected as local, and ESXi marks them as remote. When devices are not recognized as the local flash devices, they are excluded from the list of devices offered for vSAN or virtual flash resource. Marking these devices as local flash makes them available for vSAN and virtual flash resource.

Mark Storage Devices as Flash

If ESXi does not recognize its devices as flash, mark them as flash devices.

ESXi does not recognize certain devices as flash when their vendors do not support automatic flash disk detection. The Drive Type column for the devices shows HDD as their type.

Caution Marking the HDD devices as flash might deteriorate the performance of datastores and services that use them. Mark the devices only if you are certain that they are flash devices.

Prerequisites

Verify that the device is not in use.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 From the list of storage devices, select one or several HDD devices and click the **Mark as Flash Disk** (F) icon.

5 Click **Yes** to save your changes.

Results

The type of the devices changes to flash.

What to do next

If the flash device that you mark is shared among multiple hosts, make sure that you mark the device from all hosts that share the device.

Mark Storage Devices as Local

ESXi enables you to mark devices as local. This action is useful in cases when ESXi is unable to determine whether certain devices are local.

Prerequisites

- Make sure that the device is not shared.
- Power off virtual machines that reside on the device and unmount an associated datastore.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **Storage**, click **Storage Devices**.
4. From the list of storage devices, select one or several remote devices and click the **Mark as Local** icon.
5. Click **Yes** to save your changes.

About Virtual Flash Resource in vSphere Environment

You can aggregate local flash devices on an ESXi host into a single virtualized caching layer called virtual flash resource. When you set up the virtual flash resource, you create a new file system, Virtual Flash File System (VFFS). VFFS is a derivative of VMFS, which is optimized for flash devices and is used to group the physical flash devices into a single caching resource pool. As a non-persistent resource, it cannot be used to store virtual machines.

After you set up the virtual flash resource, you can use it for I/O caching filters. See Chapter 15 Filtering Virtual Machine I/O in vSphere.

Considerations for Virtual Flash Resource

When you configure a virtual flash resource, several considerations apply.

- You can have only one virtual flash resource on a single ESXi host. The virtual flash resource is managed at the host's level.
- You cannot use the virtual flash resource to store virtual machines. Virtual flash resource is a caching layer only.
- You can use only local flash devices for the virtual flash resource.
- You can create the virtual flash resource from mixed flash devices. All device types are treated equally and no distinction is made between SAS, SATA, or PCI express connectivity. When creating the resource from mixed flash devices, make sure to group similar performing devices together to maximize performance.
- You cannot use the same flash devices for the virtual flash resource and vSAN. Each requires its own exclusive and dedicated flash device.

Set Up Virtual Flash Resource

You can set up a virtual flash resource or add capacity to existing virtual flash resource.

To set up a virtual flash resource, you use local flash devices connected to your host or host cluster. To increase the capacity of your virtual flash resource, you can add more devices, up to the maximum number indicated in the *Configuration Maximums* documentation. An individual flash device must be exclusively allocated to the virtual flash resource. No other vSphere functionality, such as vSAN or VMFS, can share the device with the virtual flash resource.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Virtual Flash**, click **Virtual Flash Resource Management**.

4 Click one of the following options.

Option	Description
Add Capacity	If you are creating the virtual flash resource on an individual host.
Add Capacity on Cluster	If you are creating the virtual flash resource on a cluster.

5 From the list of available entities, select one or more to use for the virtual flash resource and click **OK**.

If your flash devices do not appear on the list, see Marking Storage Devices on ESXi Hosts.

Option	Description
Local VMware Disk	Select any combination of unclaimed flash devices. ESXi creates the VFFS volume on one of the devices and then extends the volume on the rest of the devices. The system configures the virtual flash resource on the entire VFFS volume. If a VFFS volume exists on your host, you cannot select any unclaimed devices without first selecting the existing VFFS volume.
volume ID **- Configure using the existing VFFS volume extents**	If you previously created a VFFS volume on one of the host's flash devices using the `vmkfstools` command, the volume also appears on the list of eligible entities. You can select just this volume for the virtual flash resource. Or combine it with the unclaimed devices. ESXi uses the existing VFFS volume to extend it over other devices.

Results

The virtual flash resource is created. The Device Backing area lists all devices that you use for the virtual flash resource.

What to do next

Use the virtual flash resource for I/O caching filters developed through vSphere APIs for I/O Filtering. See Using Cache I/O Filters with Flash Storage Devices on ESXi Hosts.

You can increase the capacity by adding more flash devices to the virtual flash resource.

Remove Virtual Flash Resource

You might need to remove a virtual flash resource deployed on local flash devices connected to the ESXi host. Removing the virtual flash resource frees the devices for other services.

Prerequisites

- Verify that the virtual flash resource is not used for I/O filters.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Virtual Flash**, click **Virtual Flash Resource Management** and click **Remove All**.

Results

After you remove the virtual flash resource and erase the flash device, the device is available for other operations.

Set Alarm for Virtual Flash Use

Set an alarm to indicate when the use of a virtual flash resource on your ESXi host exceeds specified threshold.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 Select the setting to change and click the **Edit** button.

Parameter	Description
VFLASH.ResourceUsageThreshold	The system triggers the `Host vFlash resource usage` alarm when a virtual flash resource use exceeds the threshold. The default threshold is 80%. You can change the threshold to an appropriate value. The alarm is cleared when the virtual flash resource use drops below the threshold.

5 Click **OK**.

Configure Host Cache with VMFS Datastore in vSphere Environment

Enable your ESXi host to swap to the host cache. You can also change the amount of space allocated for the host cache.

Your ESXi hosts can use a portion of a flash-backed storage entity as a swap cache shared by all virtual machines.

The host-level cache is made up of files on a low-latency disk that ESXi uses as a write-back cache for virtual machine swap files. All virtual machines running on the host share the cache. Host-level swapping of virtual machine pages makes the best use of potentially limited flash device space.

Prerequisites

Create a VMFS datastore using flash devices as backing. See Create a vSphere VMFS Datastore.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Host Cache Configuration**.

4 Select the flash datastore in the list and click the **Edit** icon.

5 Allocate appropriate space for host cache.

6 Click **OK**.

About VMware NVMe Storage

8

Non-Volatile Memory (NVM) storage devices that use persistent memory have become popular in data centers. To connect to local and networked NVM devices, ESXi uses the NVM Express (NVMe) protocol, a standardized protocol designed specifically for high-performance multi-queue communication with NVM devices.

VMware NVMe Concepts

Before you begin working with NVMe storage in the ESXi environment, learn about basic NVMe concepts.

NVM Express (NVMe)

NVMe is a method for connecting and transferring data between a host and a target storage system. NVMe is designed for use with faster storage media equipped with non-volatile memory, such as flash devices. This type of storage can achieve low latency, low CPU usage, and high performance, and generally serves as an alternative to SCSI storage.

NVMe Transports

The NVMe storage can be directly attached to a host using a PCIe interface or indirectly through different fabric transports. VMware NVMe over Fabrics (NVMe-oF) provides a distance connectivity between a host and a target storage device on a shared storage array.

The following types of transports for NVMe currently exist.

NVMe Transport	ESXi Support
NVMe over PCIe	Local storage.
NVMe over RDMA	Shared NVMe-oF storage. With the RoCE v2 technology.
NVMe over Fibre Channel	Shared NVMe-oF storage.
NVMe over TCP	Shared NVMe-oF storage.

For specific configuration requirements for each transport, see Requirements for VMware NVMe Storage.

NVMe Namespaces

In the NVMe storage array, a namespace is a storage volume backed by some quantity of non-volatile memory. In the context of ESXi, the namespace is analogous to a storage device,

or LUN. After your ESXi host discovers the NVMe namespace, a flash device that represents the namespace appears on the list of storage devices in the vSphere Client. You can use the device to create a datastore and store virtual machines.

Targets can present namespaces, equivalent to LUNs in SCSI, to a host in active/active or asymmetric access modes. ESXi can discover and use namespaces presented in either way. ESXi supports *NVMe end to end*, and *without any NVMe to SCSI* emulation.

NVMe Controllers

A controller is associated with one or several NVMe namespaces and provides an access path between the ESXi host and the namespaces in the storage array. To access the controller, the host can use two mechanisms, controller discovery and controller connection. For information, see Add Controllers for NVMe over Fabrics.

Controller Discovery

With this mechanism, the ESXi host first contacts a discovery controller. The discovery controller returns a list of available controllers. After you select a controller for your host to access, all namespaces associated with this controller become available to your host.

Controller Connection

Your ESXi host connects to the controller that you specify. All namespaces associated with this controller become available to your host.

NVMe Subsystem

Generally, an NVMe subsystem is a storage array that might include several NVMe controllers, several namespaces, a non-volatile memory storage medium, and an interface between the controller and non-volatile memory storage medium. The subsystem is identified by a subsystem NVMe Qualified Name (NQN).

VMware High-Performance Plug-in (HPP)

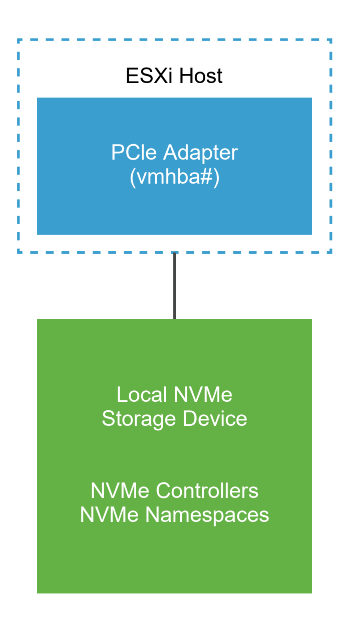
By default, the ESXi host uses the HPP to claim the NVMe-oF targets. When selecting physical paths for I/O requests, the HPP applies an appropriate Path Selection Scheme (PSS). For information about the HPP, see VMware High Performance Plug-In and Path Selection Schemes. To change the default path selection mechanism, see Change the Path Selection Policy.

Basic VMware NVMe Architecture and Components

ESXi supports local NVMe over PCIe storage and shared NVMe-oF storage, such as NVMe over Fibre Channel, NVMe over RDMA (RoCE v2), and NVMe over TCP.

VMware NVMe over PCIe

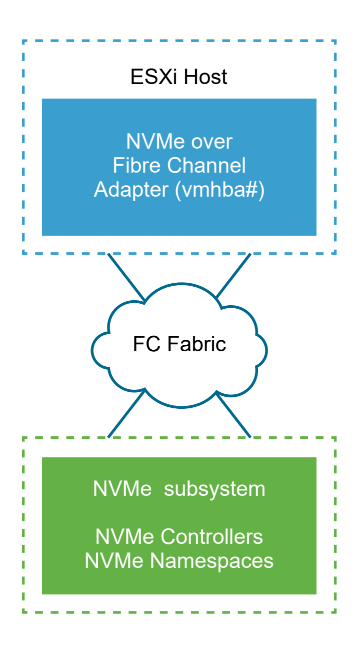
In this configuration, your ESXi host uses a PCIe storage adapter to access one or more local NVMe storage devices. After you install the adapter on the host, the host discovers available NVMe devices, and they appear in the list of storage devices in the vSphere Client.



VMware NVMe over Fibre Channel

This technology maps NVMe onto the Fibre Channel protocol to enable the transfer of data and commands between a host and a target storage device. This transport can use existing Fibre Channel infrastructure upgraded to support NVMe.

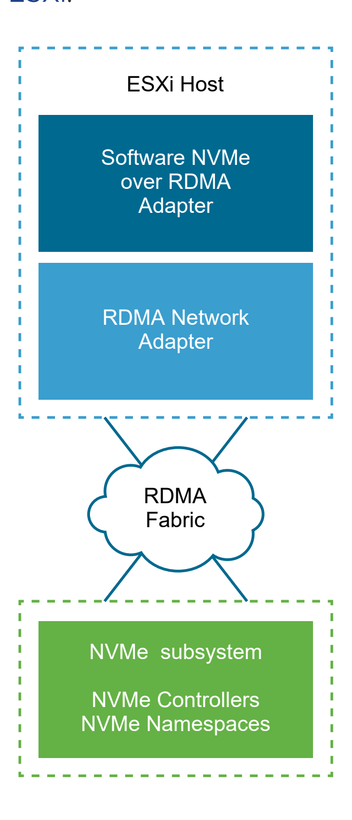
To access the NVMe over Fibre Channel storage, install a Fibre Channel storage adapter that supports NVMe on your ESXi host. You do not need to configure the adapter. It automatically connects to an appropriate NVMe subsystem and discovers all shared NVMe storage devices that it can reach. You can later reconfigure the adapter and disconnect its controllers or connect other controllers that were not available during the host boot. For more information, see Add Controllers for NVMe over Fabrics.



NVMe over RDMA (RoCE v2)

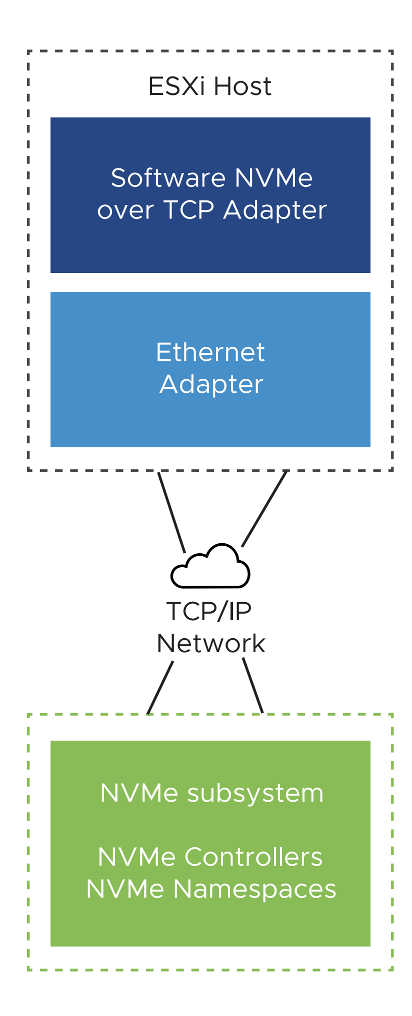
This technology uses a remote direct memory access (RDMA) transport between two systems on the network. The transport enables data exchange in the main memory bypassing the operating system or the processor of either system. ESXi supports RDMA over Converged Ethernet v2 (RoCE v2) technology, which enables a remote direct memory access over an Ethernet network.

To access storage, the ESXi host uses an RDMA network adapter installed on your host and a software NVMe over RDMA storage adapter. You must configure both adapters to use them for storage discovery. For more information, see Configuring NVMe over RDMA (RoCE v2) on ESXi.



NVMe over TCP

This technology uses Ethernet connections between two systems. To access storage, the ESXi host uses a network adapter installed on your host and a software NVMe over TCP storage adapter. You must configure both adapters to use them for storage discovery. For more information, see Configuring NVMe over TCP on ESXi.



Read the following topics next:

- Requirements for VMware NVMe Storage
- Configuring NVMe over RDMA (RoCE v2) on ESXi
- Configuring NVMe over TCP on ESXi
- Remove Software NVMe Adapters from the ESXi Host

Requirements for VMware NVMe Storage

Learn how to prepare your ESXi environment for NVMe storage. Configuration requirements might change depending on the type of NVMe transport you use. If you use NVMe over RDMA (RoCE v2), in addition to general requirements, you must also configure lossless Ethernet network.

Requirements for NVMe over PCIe

Your ESXi storage environment must include the following components:

- Local NVMe storage devices.
- Compatible ESXi host.

- Hardware NVMe over PCIe adapter. After you install the adapter, your ESXi host detects it and displays in the vSphere Client as a storage adapter (vmhba) with the protocol indicated as PCIe. You do not need to configure the adapter.

Requirements for NVMe over RDMA (RoCE v2)

- NVMe storage array with NVMe over RDMA (RoCE v2) transport support.
- Compatible ESXi host.
- Ethernet switches supporting a lossless network.
- Network adapter that supports RDMA over Converged Ethernet (RoCE v2). To configure the adapter, see Configuring NVMe over RDMA (RoCE v2) on ESXi.
- Software NVMe over RDMA adapter. This software component must be enabled on your ESXi host and connected to an appropriate network RDMA adapter. For information, see Add Software NVMe over RDMA or NVMe over TCP Adapters.
- NVMe controller. You must add a controller after you configure the software NVMe over RDMA adapter. See Add Controllers for NVMe over Fabrics.
- Lossless Ethernet. See Configuring Lossless Ethernet for NVMe over RDMA.

Requirements for NVMe over Fibre Channel

- Fibre Channel storage array that supports NVMe. For information, see Chapter 3 Using ESXi with Fibre Channel SAN.
- Compatible ESXi host.
- Hardware NVMe adapter. Typically, it is a Fibre Channel HBA that supports NVMe. When you install the adapter, your ESXi host detects it and displays in the vSphere Client as a standard Fibre Channel adapter (vmhba) with the storage protocol indicated as NVMe. You do not need to configure the hardware NVMe adapter to use it.
- NVMe controller. You do not need to configure the controller. After you install the required hardware NVMe adapter, it automatically connects to all targets and controllers that are reachable at the moment. You can later disconnect the controllers or connect other controllers that were not available during the host boot. See Add Controllers for NVMe over Fabrics.

Requirements for NVMe over TCP

- NVMe storage array with NVMe over TCP transport support.
- Compatible ESXi host.
- An Ethernet adapter. To configure the adapter, see Configuring NVMe over TCP on ESXi.
- Software NVMe over TCP adapter. This software component must be enabled on your ESXi host and connected to an appropriate network adapter. For more information, see Add Software NVMe over RDMA or NVMe over TCP Adapters.

- NVMe controller. You must add a controller after you configure the software NVMe over TCP adapter. See Add Controllers for NVMe over Fabrics.

VMware NVMe over Fabrics Shared Storage Support

In the ESXi environment, the NVMe storage devices appear similar to SCSI storage devices, and can be used as shared storage. Follow these rules when using the NVMe-oF storage.

- Do not mix transport types to access the same namespace.
- Make sure that active paths are presented to the host. The namespaces cannot be registered until the active path is discovered.

Shared Storage Functionality	SCSI over Fabric Storage	NVMe over Fabric Storage
RDM	Supported	Not supported
Core dump	Supported	Supported
SCSI-2 reservations	Supported	Not supported
Clustered VMDK	Supported	Supported
Shared VMDK with multi-writer flag	Supported	Supported In vSphere 7.0 Update 1 and later. For more information, see the Knowledge Base article.
Virtual Volumes	Supported	Supported In vSphere 8.0 and later. For more information, see NVMe and Virtual Volumes in vSphere.
Hardware acceleration with VAAI plug-ins	Supported	Not supported
Default MPP	NMP	HPP (NVMe-oF targets cannot be claimed by NMP)

Configuring Lossless Ethernet for NVMe over RDMA

NVMe over RDMA in ESXi requires lossless Ethernet network.

To establish lossless networks, you can select one of the available QoS settings.

Enable Global Pause Flow Control

In this network configuration, ensure global pause flow control is enabled on the network switch ports. Also, ensure that RDMA capable NICs in the host auto-negotiate to the correct flow control automatically.

To check flow control, run the following command:

```
#esxcli network nic get -n vmnicX
   Pause RX: true
   Pause TX: true
```

If the above command options are not set to true, run the following command.

```
#esxcli network nic pauseParams set -r true -t true -n vmnicX
```

Enable Priority Flow Control

For RoCE traffic to be lossless, you must configure the PFC priority value to 3 in the physical switch and hosts. You can configure the PFC in the ESXi host in two ways:

- Automatic Configuration. Apply DCB PFC configuration automatically on the host RNIC, if the RNIC driver supports DCB and DCBx.

  You can verify the current DCB settings by running the following command:

  ```
  #esxcli network nic dcb status get -n vmnicX
  ```

- Manual configuration. In some cases, the RNIC drivers provide a method to manually configure the DCB PFC using driver specific parameters. To use this method, see vendor specific driver documentation. For example, in Mellanox ConnectX-4/5 driver, you can set the PFC priority value to three by running the following command and then rebooting the host.

  ```
  #esxcli system module parameters set -m nmlx5_core -p "pfctx=0x08 pfcrx=0x08"
  ```

Enable DSCP based PFC

DSCP based PFC is another way to configure lossless network. In physical switches and hosts, you must set the DSCP value to 26. To use this option, see vendor specific driver documentation. For example, in Mellanox ConnectX-4/5 driver, you can set the DSCP tag value to 26 by running the following commands.

1. Enable PFC and DSCP trust mode.

   ```
   #esxcli system module parameters set -m nmlx5_core -p "pfctx=0x08 pfcrx=0x08 trust_state=2"
   ```

2. Set DSCP value to 26.

   ```
   #esxcli system module parameters set -m nmlx5_rdma -p "dscp_force=26"
   ```

3. Verify parameters to check to confirm if settings are correct and set.

   ```
   esxcli system module parameters list -m nmlx5_core | grep 'trust_state\|pfcrx\|pfctx'
   ```

4. Reboot the host.

Configuring NVMe over RDMA (RoCE v2) on ESXi

The adapter configuration process on the ESXi host involves setting up VMkernel binding for an RDMA network adapter, and then adding a software NVMe over RDMA adapter. You can then add an NVMe controller.

The entire configuration process includes these actions.

Action	Description
View RDMA Network Adapters	On your ESXi host, install a network adapter that supports RDMA (RoCE v2). For example, Mellanox Technologies MT27700 Family ConnectX-4. After you install the network adapter, use the vSphere Client to review the RDMA adapter and a physical network adapter.
Configure VMkernel Binding for the RDMA Adapter	Port binding for NVMe over RDMA involves creating a switch and connecting the physical network adapter and the VMkernel adapter to the switch. Through this connection, the RDMA adapter becomes bound to the VMkernel adapter. In the configuration, you can use a vSphere standard switch or a vSphere distributed switch.
Add Software NVMe over RDMA or NVMe over TCP Adapters	Use the vSphere Client to activate the software storage adapters for NVMe over RDMA.
Add Controllers for NVMe over Fabrics	Use the vSphere Client to add an NVMe controller. After you add the controller, the NVMe namespaces associated with the controller become available to your ESXi host. The NVMe storage devices that represent the namespaces in the ESXi environment appear on the storage devices list.

The following video walks you through the steps of configuring NVMe over RDMA adapters.



(Setting up NVMe over RDMA Adapters)

View RDMA Network Adapters

After you install a network adapter that supports RDMA (RoCE v2) on your ESXi host, use the vSphere Client to review the RDMA adapter and a physical network adapter.

Procedure

1 On your ESXi host, install an adapter that supports RDMA (RoCE v2), for example, Mellanox Technologies MT27700 Family ConnectX-4.

  The host discovers the adapter and the vSphere Client displays its two components, an RDMA adapter and a physical network adapter.

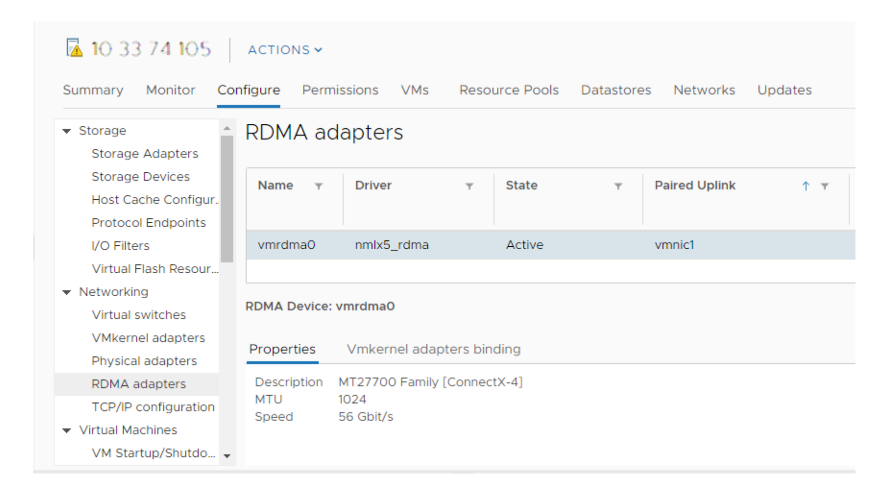
2 In the vSphere Client, verify that the RDMA adapter is discovered by your host.

  a Navigate to the host.

  b Click the **Configure** tab.

c Under **Networking**, click **RDMA adapters**.

In this example, the RDMA adapter appears on the list as `vmrdma0`. The **Paired Uplink** column displays the network component as the `vmnic1` physical network adapter.

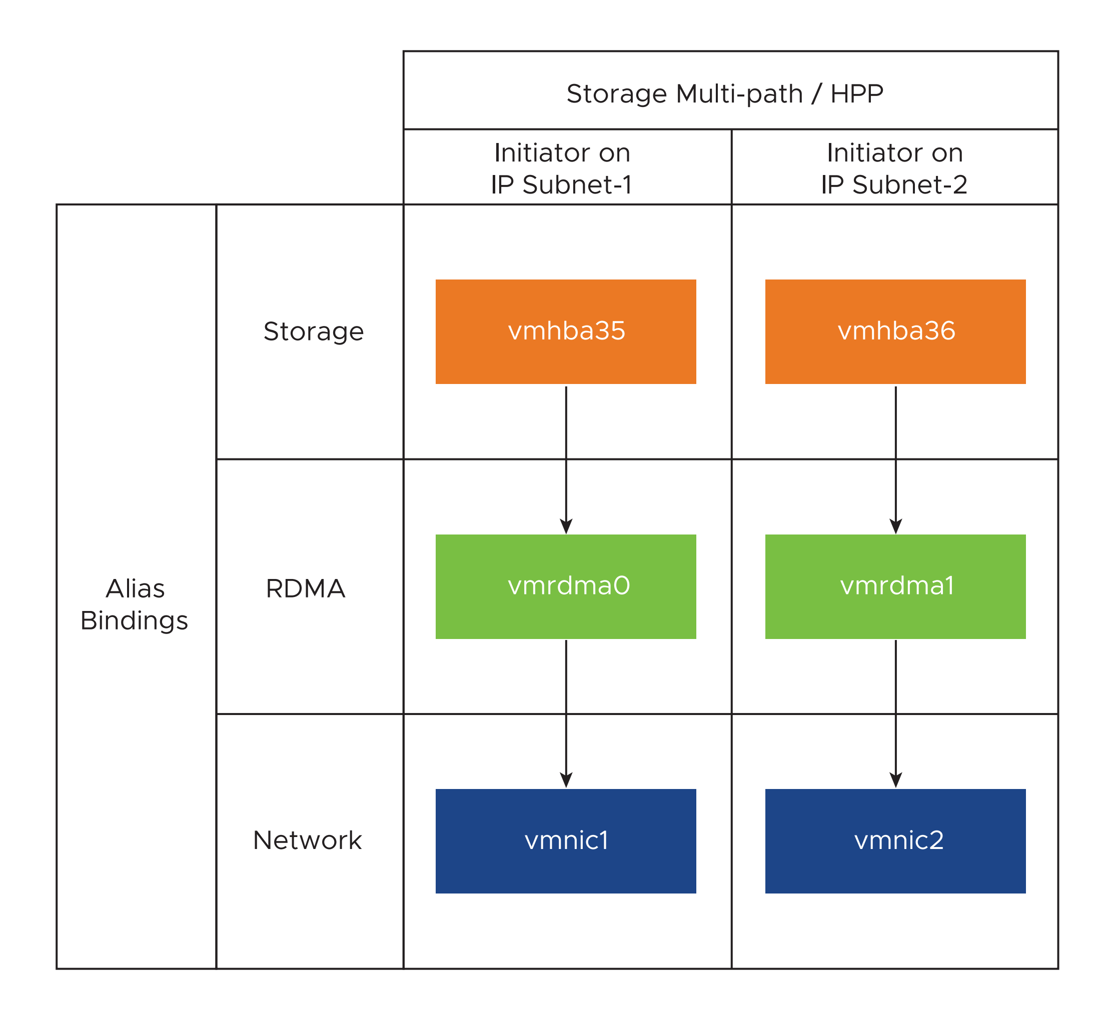


d To verify the description of the adapter, select the RDMA adapter from the list, and click the **Properties** tab.

Configure VMkernel Binding for the RDMA Adapter

Port binding for NVMe over RDMA involves creating a switch and connecting the physical network adapter and the VMkernel adapter to the switch. Through this connection, the RDMA adapter becomes bound to the VMkernel adapter. In the configuration, you can use a vSphere standard switch or a vSphere distributed switch.

The following diagram displays the port binding for the NVMe over RDMA adapter.



For more information about creating switches, see *Create a vSphere Standard Switch* or *Create a vSphere Distributed Switch* in the *vSphere Networking* documentation.

Example of Network Topology with NVMe over RDMA

In this example, two vSphere standard switches and two uplinks (RDMA capable NICs) provide high availability. They connect to two controller pairs in two subnets.

HA with Multiple vSwitches and Multiple Uplinks (RNICs)

ESXi
Paths
4 Paths= 2 NICs (\w 2 vSwitches) x 2 Controller pairs (in 2 Subnets)
vmhba
vmhba1
vmhba2
portgroup/
vmknic
vmk1
vmk2
PG1
192.168.50x
vSwitch 1
PG2
192.168.51.x
vSwitch 2
vmnic/
vmrdma
vmrdma0
vmnic1
vmnic2
vmrdma1
External Switch 1
External Switch 2
Subnet-0
Subnet-1
Subnet-0
Subnet-1
C0
192.168.50.100
C1
192.168.51.100
C0
192.168.50.200
C1
192.168.51.200
TARGET
NVME- FABRICS (RDMA)

Configure VMkernel Binding with a vSphere Standard Switch

You can configure VMkernel port binding for the RDMA adapter using a vSphere standard switch and one uplink per switch. Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You use 1:1 mapping between each virtual and physical network adapter.

Procedure

1 Create a vSphere standard switch with a VMkernel adapter and the network component.

   a In the vSphere Client, select your host and click the **Networks** tab.

   b Click **Actions > Add Networking**.

   c Select **VMkernel Network Adapter** and click **NEXT**.

   d Select **New standard switch** and click **NEXT**.

   e Under **Assigned adapters**, click **+**.

   The list of available physical adapters is displayed.

   f Select the required physical adapter `vmnic`, and click **OK**.

   Note Ensure to select the physical network adapter that corresponds to the RDMA adapter. To see the association between the RDMA adapter `vmrdma`, and the physical network adapter `vmnic`, see View RDMA Network Adapters.

   ---

   g Under **VMkernel port settings**, enter the required values.

   If you are using VLAN for the storage path, enter the VLAN ID.

   h In the **IP settings** list, enter the VMkernel IPv4 settings.

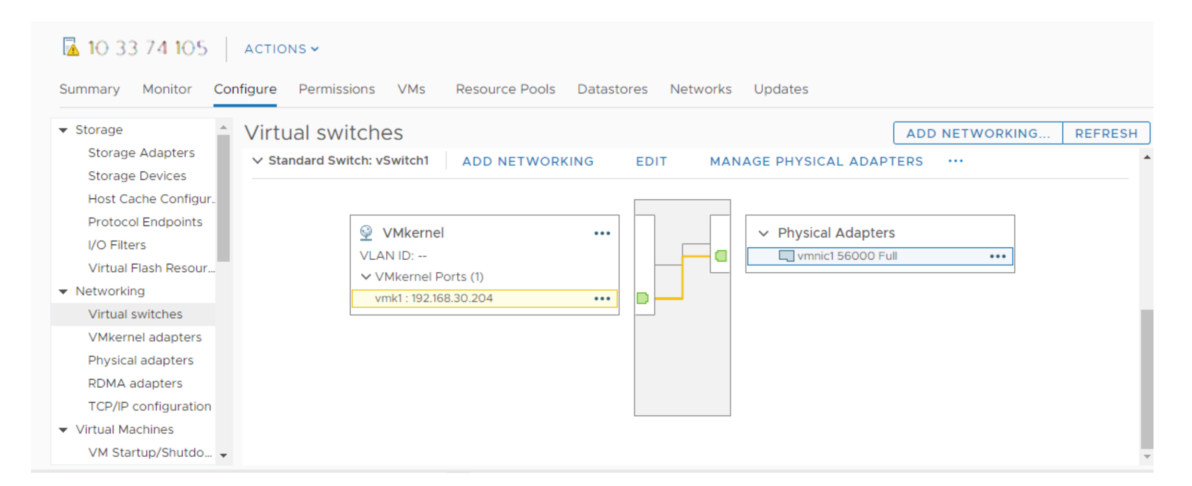
   i Under Available services, select **NVMe over RDMA**.

2 Verify that your switch is correctly configured.

   a On the **Configure** tab, select **Virtual switches** under **Networking**.

   b Expand the switch and verify its configuration.

   The illustration shows that the physical network adapter and the VMkernel adapter are connected to the vSphere standard switch. Through this connection, the RDMA adapter is bound to the VMkernel adapter.

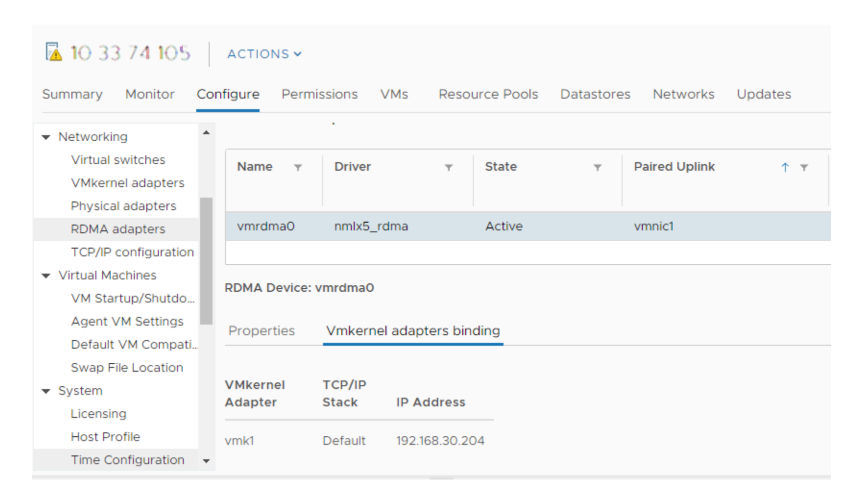


3 Verify the configuration of the VMkernel binding for the RDMA adapter.

   a Under **Networking** list, click **RDMA adapters**, and select the RDMA adapter from the list.

   b Click the **VMkernel adapters binding** tab and verify that the associated VMkernel adapter appears on the page.

   In this example, the `vmrdma0` RDMA adapter is paired to the `vmnic1` network adapter and is connected to the `vmk1` VMkernel adapter.



Configure VMkernel Binding with a vSphere Standard Switch and NIC Teaming

You can configure VMkernel port binding for the RDMA adapter using a vSphere standard switch with the NIC teaming configuration. You can use NIC teaming to achieve network redundancy. You can configure two or more network adapters (NICs) as a team for high availability and load balancing.

Procedure

1 Create a vSphere standard switch with a VMkernel adapter and the network component with the NIC teaming configuration.

   a In the vSphere Client, select your host and click **Networks** tab.

   b Click **Actions > Add Networking**.

   c Select **VMkernel Network Adapter** and click **NEXT**.

   d Select **New standard switch** and click **NEXT**.

   e Under **Assigned adapters**, click **+**.

   A list of available physical adapters is displayed.

   f Select the required physical adapter `vmnic`, and add it under **Active adapters**.

   g Select another physical adapter `vmnic`, and add it under **Unused adapters**.

   h Under **VMkernel port settings**, enter the required values.

   If you are using VLAN for the storage path, enter the VLAN ID.

i In the **IP settings** list, specify VMkernel IPv4 settings.

j Under Available services, select **NVMe over RDMA**.

Repeat step 1 to configure an existing standard switch.

2 Configure your switch for NIC teaming configuration.

a Click the **Configure** tab, and select **Virtual switches** under **Networking**.

b Select the appropriate VMkernel adapter.

c From the right-click menu, click **Edit Settings**.

d Select **Teaming and Failover**.

e Under **Active adapters**, move the required physical adapter `vmnic`.

f Under **Standby adapters > Failover order**, move the other physcial adapters.

g Set appropriate load balancing and other properties.

h Repeat the steps to configure additional VMkernel adapters.

3 Repeat steps 1 and 2 to add and configure additional set of teamed `rnic`s. To verify if the adapter is configured, click the **Configure** tab and select **VMkernel adapters**

Configure VMkernel Binding with a vSphere Distributed Switch

You can configure VMkernel port binding for the RDMA adapter using a vSphere distributed switch and one uplink per switch. Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You use 1:1 mapping between each virtual and physical network adapter.

Procedure

1 Create a vSphere distributed switch with a VMkernel adapter and the network component.

a In the vSphere Client, select **Datacenter**, and click the **Networks** tab.

b Click **Actions** , and select **Distributed Switch > New Distributed Switch**.

c Select a name for the switch.

Ensure that the location of the data center is present within your host, and click **Next**.

d Select a compatible ESXi version, and click **Next**.

e Enter the required number of uplinks, and click **Finish**.

2 Add one or more hosts to your distributed virtual switch.

a In the vSphere Client, select **Datacenter**, and click **Distributed Switches**..

A list of available DSwitches appear.

b Right-click the DSwitch, and select **Add and Manage Hosts** from the menu.

c Select **Add hosts**, and click **Next**.

d Select your host, and click **Next**.

e Select **Assign uplink**.

f Enter the relevant uplink to assign the `vmnic`.

g Assign a VMkernel adapter, and click **Next**.

h In the vSphere Client, select the DSwitch, and click the **Ports** tab.

You can view the uplinks created for your switch here.

3 Create distributed port groups for the NVMe over RDMA storage path.

a In the vSphere Client, select the required DSwitch.

b Click **Actions** and select **Distributed Port Group > New Distributed Port Group**.

c Under **Configure Settings**, enter the general properties of the port group.

If you have configured a specific VLAN, add it in the VLAN ID.

Note Network connectivity issues might occur if you do not configure VLAN properly.

4 Configure the VMkernel adapters.

a In the vSphere Client, expand the **DSwitch** list, and select the distributed port group.

b Click **Actions > Add VMkernel Adapters**.

c In the **Select Member Hosts** dialog box, select your host and click **OK**.

d In the **Configure VMkernel Adapter** dialog box, ensure that the MTU matches to the Switch MTU.

e Under **Available services**, select **NVMe over RDMA** for appropriate tagging.

f Click **Finish**.

g Repeat step b and step c to add multiple RDMA capable NICs.

5 Set NIC teaming policies for the distributed port groups.

a In the **Distributed Port Group**, click **Actions > Edit Settings**.

b Click **Teaming and Failover**, and verify the active uplinks.

c Assign one uplink as **Active** for the port group, and the other uplink as **Unused**.

Repeat step c for each of the port groups created.

What to do next

After you complete the configuration, click **Configure**, and verify whether the physical adapter tab on your host lists the DVSwitch for the NICs selected.

Add Software NVMe over RDMA or NVMe over TCP Adapters

ESXi supports NVMe over RDMA and NVMe over TCP software adapters. Use the vSphere Client to add the software storage adapters for NVMe over RDMA or NVMe over TCP.

Prerequisites

- On your ESXi host, install an adapter that supports the following types of storage.
  - NVMe over RDMA adapter. For example, Mellanox Technologies MT27700 Family ConnectX-4.
  - NVMe over TCP adapter. For example, i40en.
- Configure the VMkernel binding for your adapters.
  - For NVMe over RDMA, see Configure VMkernel Binding for the RDMA Adapter.
  - For NVMe over TCP, see Configure VMkernel Binding for the NVMe over TCP Adapter.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and click the **Add Software Adapter** icon.

4 Select the adapter type as required.

- **NVMe over RDMA adapter**
- **NVMe over TCP adapter**

5 Depending on your selection in Step 4, select an appropriate RDMA adapter or TCP network adapter (`vmnic`) from the drop-down menu.

> **Note** If you get an error message that prevents you from creating the software adapter, make sure that the VMkernel binding for the adapter is configured correctly. For more information, see Configure VMkernel Binding for the RDMA Adapter and Configure VMkernel Binding for the NVMe over TCP Adapter.

Results

The software NVMe over RDMA and NVMe over TCP adapters appear in the list as `vmhba` storage adapters. You can remove the adapters if you need to free the underlying RDMA and TCP network adapter for other purposes. See Remove Software NVMe Adapters from the ESXi Host.

Add Controllers for NVMe over Fabrics

Use the vSphere Client to add an NVMe controller. After you add the controller, the NVMe namespaces associated with the controller become available to your ESXi host. The NVMe

storage devices that represent the namespaces in the ESXi environment appear on the storage devices list.

Prerequisites

- Make sure that your ESXi host has appropriate adapters for your type of storage. See Requirements for VMware NVMe Storage.
- If you use NVMe over RDMA (RoCE v2) storage, perform these tasks:
  - Configure VMkernel Binding for the RDMA Adapter
  - Add Software NVMe over RDMA or NVMe over TCP Adapters
- If you use NVMe over TCP storage, perform these tasks:
  - Configure VMkernel Binding for the NVMe over TCP Adapter
  - Add Software NVMe over RDMA or NVMe over TCP Adapters

Note With NVMe over Fibre Channel, after you install the required adapter, it automatically connects to all targets that are reachable at the moment. You can later reconfigure the adapter and disconnect its controllers or connect other controllers that were not available during the host boot.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.
4. Click the **Controllers** tab, and click **Add Controller**.

5 On the **Add controller** dialog box, select one of the following discovery methods.

Option	Description
Automatically	This option indicates that your host can discover controllers automatically and accept connection to any available controller. a Specify the following parameters for to discover controllers. ■ For NVMe over RDMA (RoCE v2), the IP address and transport port number. ■ For NVMe over TCP, the IP address, transport port number, and the digest parameter. b Click **Discover Controllers**. c From the list of controllers, select the controller to use.
Manually	With this method, you manually enter controller details. The host requests a connection to a specific controller using the parameters you specify: ■ Subsystem NQN ■ Target port identification. ■ For NVMe over RDMA (RoCE v2), the IP address and transport port number (optional). ■ For NVMe over TCP, the IP address, transport port number (optional), and the digest parameter (optional). ■ For NVMe over Fibre Channel, the WorldWideNodeName and WorldWidePortName. ■ Admin queue size. An optional parameter that specifies the size of the admin queue of the controller. A default value is 16. ■ Keepalive timeout. An optional parameter to specify in seconds the keep alive timeout between the adapter and the controller. A default timeout value is 60 seconds. **Note** IO Queue Size and IO Queue Number are optional parameters that can be set only through `esxcli`.

Results

The controller appears on the list of controllers. Your host can now discover the NVMe namespaces that are associated with the controller. The NVMe storage devices that represent the namespaces in the ESXi environment appear on the storage devices list in the vSphere Client.

Configuring NVMe over TCP on ESXi

The adapter configuration process on the ESXi host involves setting up VMkernel binding for a TCP network adapter, and then adding a software adapter for NVMe over TCP. After that, you can add an NVMe controller.

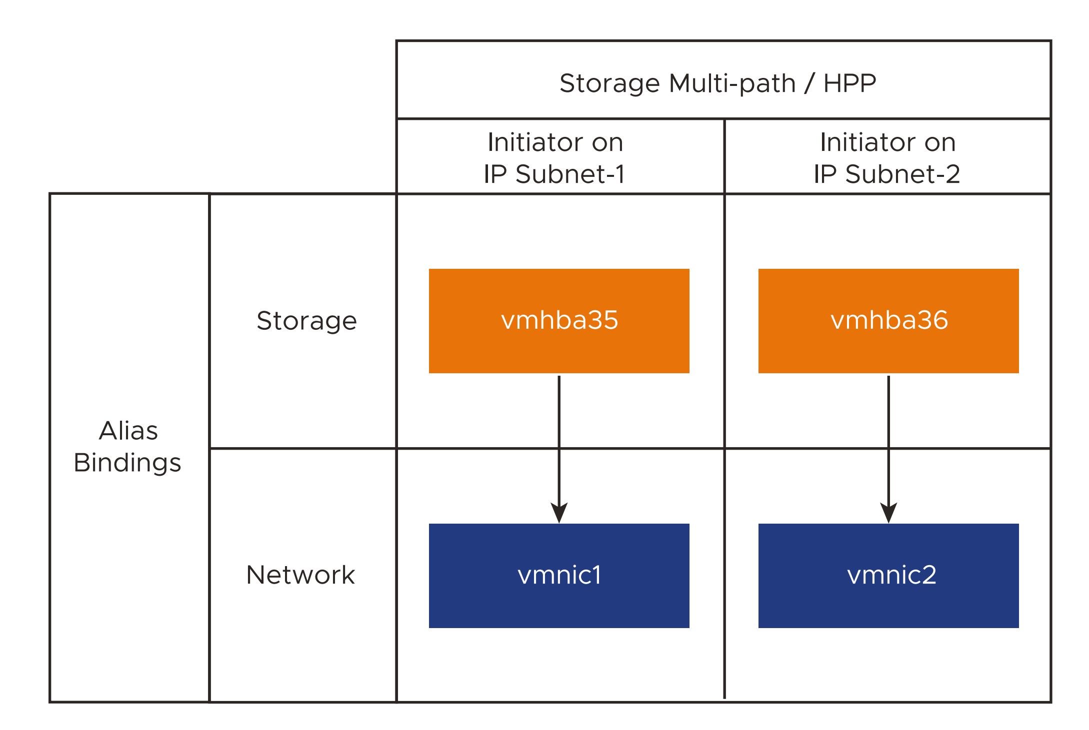
The entire configuration process includes these actions.

Action	Description
On your ESXi host, install an adapter that supports NVMe over TCP technology	For example, i40en.
Configure VMkernel Binding for the NVMe over TCP Adapter	VMkernel binding for NVMe over TCP involves creating a virtual switch and connecting the physical network adapter and the VMkernel adapter to the virtual switch. Through this connection, the TCP adapter becomes bound to the VMkernel adapter. In the configuration, you can use a vSphere standard switch or a vSphere distributed switch.
Add Software NVMe over RDMA or NVMe over TCP Adapters	Use the vSphere Client to enable the software storage adapters for NVMe over TCP.
Add Controllers for NVMe over Fabrics	Use the vSphere Client to add an NVMe controller. After you add the controller, the NVMe namespaces associated with the controller become available to your ESXi host. The NVMe storage devices that represent the namespaces in the ESXi environment appear on the storage devices list.

Configure VMkernel Binding for the NVMe over TCP Adapter

Port binding for NVMe over TCP involves creating a virtual switch and connecting the physical network adapter and the VMkernel adapter to the virtual switch. Through this connection, the TCP adapter becomes bound to the VMkernel adapter. In the configuration, you can use a vSphere standard switch or a vSphere distributed switch.

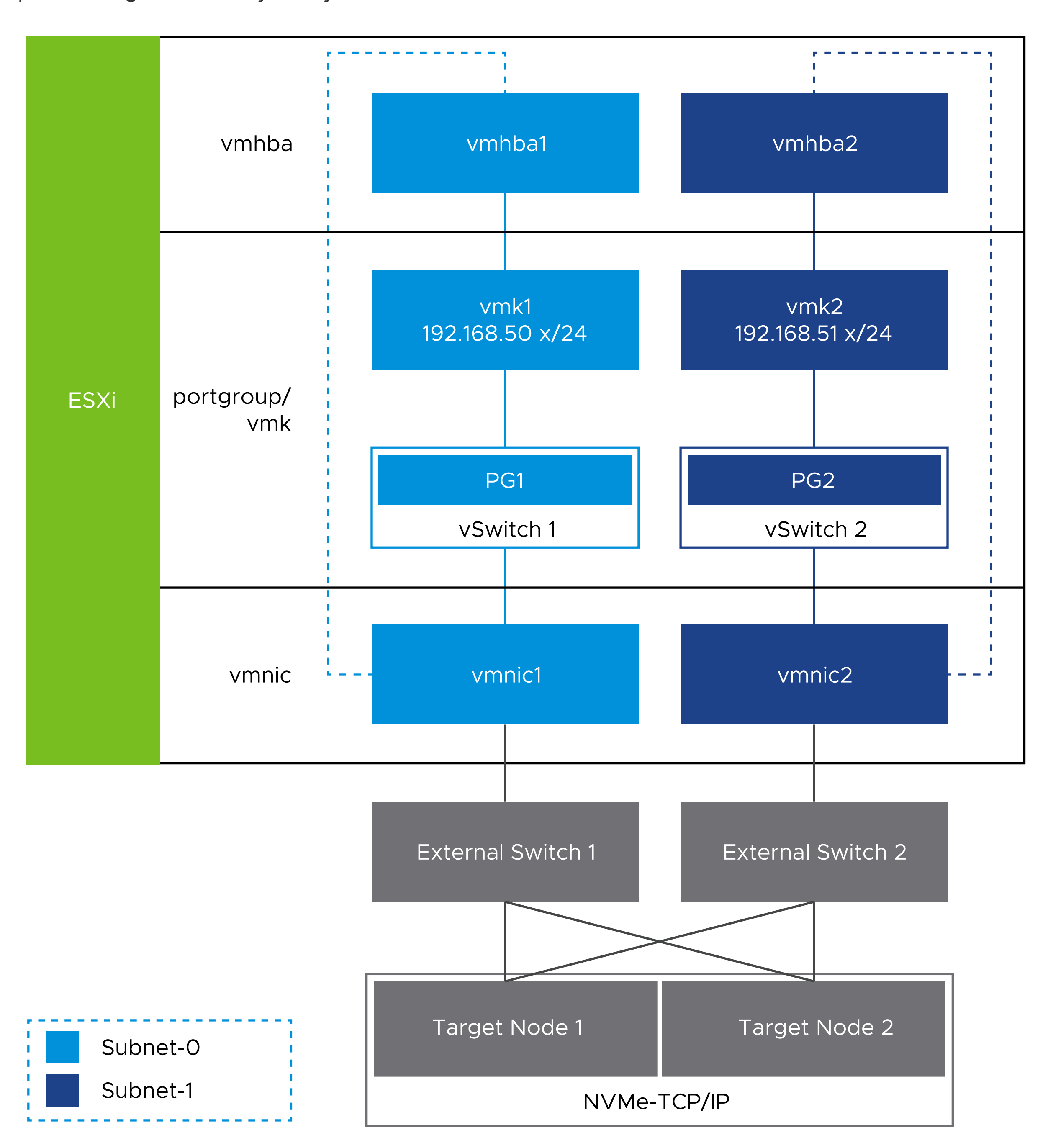
The following diagram displays the port binding for the NVMe over TCP adapter.



For more information about creating switches, see *Create a vSphere Standard Switch* or *Create a vSphere Distributed Switch* in the *vSphere Networking* documentation.

Example of Network Topology with NVMe over TCP

In this example, two vSphere standard switches and two network adapters (vmnic) on the host provide high availability. They connect to two external switches.



Configure VMkernel Binding for the TCP Adapter with a vSphere Standard Switch

You can configure VMkernel binding for the TCP adapter using a vSphere standard switch and one uplink per switch. Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You use 1:1 mapping between each virtual and physical network adapter.

Procedure

1 Create a vSphere standard switch with a VMkernel adapter and the network component.

   a In the vSphere Client, select your host and click the **Networks** tab.

   b Click **Actions > Add Networking**.

   c Select **VMkernel Network Adapter** and click **NEXT**.

   d Select **New standard switch** and click **NEXT**.

   e Under **Assigned adapters**, click **+**.

   The list of available physical adapters is displayed.

   f Select the required physical adapter `vmnic`, and click **OK**.

   Note Make sure to select the physical network adapter that corresponds to the TCP/IP adapter.

   g Under **VMkernel port settings**, enter the required values.

   If you are using VLAN for the storage path, enter the VLAN ID.

   h In the **IP settings** list, enter the VMkernel IPv4 settings.

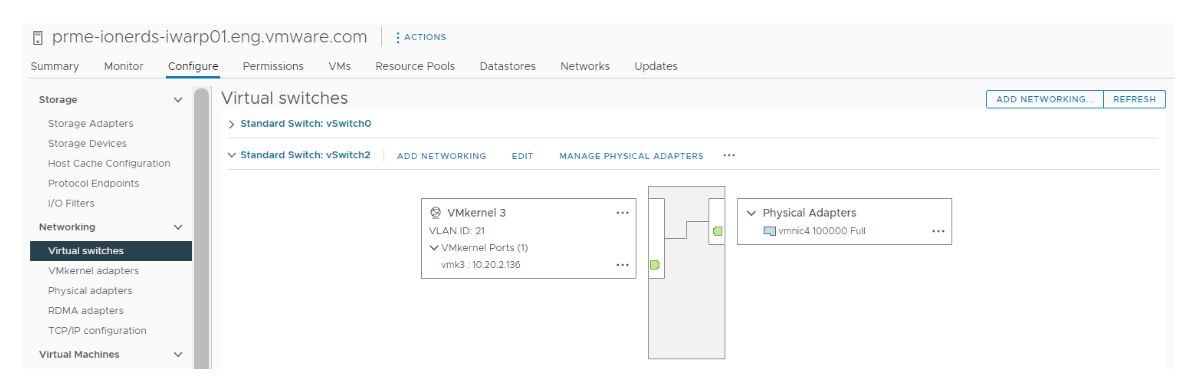
   i Under **Available services**, select **NVMe over TCP** for appropriate tagging.

2 Verify that your switch is correctly configured.

   a On the **Configure** tab, select **Virtual switches** under **Networking**.

   b Expand the switch and verify its configuration.

   The illustration shows that the physical network adapter and the VMkernel adapter are connected to the vSphere standard switch. Through this connection, the TCP adapter is bound to the VMkernel adapter.



3 Set NIC teaming policies for vSphere standard switch.

Note The NVMe over TCP adapter does not support such NIC teaming features as failover and load balancing. Instead, it relies on Storage Multipathing for these functionalities. However, if you must configure NIC teaming for other network workloads on the uplink serving the NVMe over TCP adapter, follow these steps.

   a Click the **Configure** tab, and select **Virtual switches** under **Networking**.
   b Select the appropriate VMkernel adapter.
   c From the right-click menu, click **Edit Settings**.
   d Select **Teaming and Failover**.
   e Under **Active adapters**, move the required physical adapter `vmnic`.
   f Under **Standby adapters > Failover order**, move the other physcial adapters.
   g Set appropriate load balancing and other properties.
   h Repeat the steps to configure additional VMkernel adapters.

   To verify if the adapter is configured, click the **Configure** tab and select **VMkernel adapters**.

Configure VMkernel Binding for the TCP Adapter with a vSphere Distributed Switch

You can configure VMkernel port binding for the TCP adapter using a vSphere distributed switch and one uplink per switch. Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You use 1:1 mapping between each virtual and physical network adapter.

Procedure

1 Create a vSphere distributed switch with a VMkernel adapter and the network component.
   a In the vSphere Client, select **Datacenter**, and click the **Networks** tab.
   b Click **Actions** , and select **Distributed Switch > New Distributed Switch**.
   c Select a name for the switch.

      Ensure that the location of the data center is present within your host, and click **Next**.
   d Select a compatible ESXi version, and click **Next**.
   e Enter the required number of uplinks, and click **Finish**.

2 Add one or more hosts to your distributed virtual switch.
   a In the vSphere Client, select **Datacenter**, and click **Distributed Switches**.

      A list of available DSwitches appear.
   b Right-click the DSwitch, and select **Add and Manage Hosts** from the menu.
   c Select **Add hosts**, and click **Next**.

d Select your host, and click **Next**.

e Select **Assign uplink**.

f Enter the relevant uplink to assign the `vmnic`.

g Assign a VMkernel adapter, and click **Next**.

h In the vSphere Client, select the DSwitch, and click the **Ports** tab.

You can view the uplinks created for your switch here.

3 Create distributed port groups for the NVMe over TCP storage path.

a In the vSphere Client, select the required DSwitch.

b Click **Actions** and select **Distributed Port Group > New Distributed Port Group**.

c Under **Configure Settings**, enter the general properties of the port group.

If you have configured a specific VLAN, add it in the VLAN ID.

Note Network connectivity issues might occur if you do not configure VLAN properly.

4 Configure the VMkernel adapters.

a In the vSphere Client, expand the **DSwitch** list, and select the distributed port group.

b Click **Actions > Add VMkernel Adapters**.

c In the **Select Member Hosts** dialog box, select your host and click **OK**.

d In the **Configure VMkernel Adapter** dialog box, ensure that the MTU matches to the Switch MTU.

e Click **Finish**.

f Repeat step b and step c to add multiple TCP capable NICs.

5 Set NIC teaming policies for the distributed port groups.

Note The NVMe over TCP adapter does not support such NIC teaming features as failover and load balancing. Instead, it relies on Storage Multipathing for these functionalities. However, if you must configure NIC teaming for other network workloads on the uplink serving the NVMe over TCP adapter, follow these steps.

a In the **Distributed Port Group**, click **Actions > Edit Settings**.

b Click **Teaming and Failover**, and verify the active uplinks.

c Assign one uplink as **Active** for the port group, and the other uplink as **Unused**.

Repeat step c for each of the port groups created.

What to do next

After you complete the configuration, click **Configure**, and verify whether the physical adapter tab on your host lists the DVSwitch for the NICs selected.

Add Software NVMe over RDMA or NVMe over TCP Adapters

ESXi supports NVMe over RDMA and NVMe over TCP software adapters. Use the vSphere Client to add the software storage adapters for NVMe over RDMA or NVMe over TCP.

Prerequisites

- On your ESXi host, install an adapter that supports the following types of storage.
  - NVMe over RDMA adapter. For example, Mellanox Technologies MT27700 Family ConnectX-4.
  - NVMe over TCP adapter. For example, i40en.
- Configure the VMkernel binding for your adapters.
  - For NVMe over RDMA, see Configure VMkernel Binding for the RDMA Adapter.
  - For NVMe over TCP, see Configure VMkernel Binding for the NVMe over TCP Adapter.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and click the **Add Software Adapter** icon.

4 Select the adapter type as required.
   - **NVMe over RDMA adapter**
   - **NVMe over TCP adapter**

5 Depending on your selection in Step 4, select an appropriate RDMA adapter or TCP network adapter (`vmnic`) from the drop-down menu.

   Note If you get an error message that prevents you from creating the software adapter, make sure that the VMkernel binding for the adapter is configured correctly. For more information, see Configure VMkernel Binding for the RDMA Adapter and Configure VMkernel Binding for the NVMe over TCP Adapter.

Results

The software NVMe over RDMA and NVMe over TCP adapters appear in the list as `vmhba` storage adapters. You can remove the adapters if you need to free the underlying RDMA and TCP network adapter for other purposes. See Remove Software NVMe Adapters from the ESXi Host.

Add Controllers for NVMe over Fabrics

Use the vSphere Client to add an NVMe controller. After you add the controller, the NVMe namespaces associated with the controller become available to your ESXi host. The NVMe

storage devices that represent the namespaces in the ESXi environment appear on the storage devices list.

Prerequisites

- Make sure that your ESXi host has appropriate adapters for your type of storage. See Requirements for VMware NVMe Storage.
- If you use NVMe over RDMA (RoCE v2) storage, perform these tasks:
  - Configure VMkernel Binding for the RDMA Adapter
  - Add Software NVMe over RDMA or NVMe over TCP Adapters
- If you use NVMe over TCP storage, perform these tasks:
  - Configure VMkernel Binding for the NVMe over TCP Adapter
  - Add Software NVMe over RDMA or NVMe over TCP Adapters

Note With NVMe over Fibre Channel, after you install the required adapter, it automatically connects to all targets that are reachable at the moment. You can later reconfigure the adapter and disconnect its controllers or connect other controllers that were not available during the host boot.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.
4. Click the **Controllers** tab, and click **Add Controller**.

5 On the **Add controller** dialog box, select one of the following discovery methods.

Option	Description
Automatically	This option indicates that your host can discover controllers automatically and accept connection to any available controller. a Specify the following parameters for to discover controllers. ■ For NVMe over RDMA (RoCE v2), the IP address and transport port number. ■ For NVMe over TCP, the IP address, transport port number, and the digest parameter. b Click **Discover Controllers**. c From the list of controllers, select the controller to use.
Manually	With this method, you manually enter controller details. The host requests a connection to a specific controller using the parameters you specify: ■ Subsystem NQN ■ Target port identification. ■ For NVMe over RDMA (RoCE v2), the IP address and transport port number (optional). ■ For NVMe over TCP, the IP address, transport port number (optional), and the digest parameter (optional). ■ For NVMe over Fibre Channel, the WorldWideNodeName and WorldWidePortName. ■ Admin queue size. An optional parameter that specifies the size of the admin queue of the controller. A default value is 16. ■ Keepalive timeout. An optional parameter to specify in seconds the keep alive timeout between the adapter and the controller. A default timeout value is 60 seconds. **Note** IO Queue Size and IO Queue Number are optional parameters that can be set only through `esxcli`.

Results

The controller appears on the list of controllers. Your host can now discover the NVMe namespaces that are associated with the controller. The NVMe storage devices that represent the namespaces in the ESXi environment appear on the storage devices list in the vSphere Client.

Remove Software NVMe Adapters from the ESXi Host

Use the vSphere Client to remove software NVMe over RDMA or NVMe over TCP adapters from the ESXi host. You can remove the adapter if you need to free the underlying RDMA adapter or ethernet adapter for other purposes.

You cannot remove the NVMe over PCIe and NVMe over Fibre Channel adapters.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**, and select the adapter (`vmhba#`) to remove.

4 Remove the NVMe controller connected to the adapter.

   a Click the **Controllers** tab.

   b Select the controller and click **Remove**.

      The NVMe controller is disconnected and disappears from the list.

5 Click the **Remove** icon (Remove the host's storage adapter) to remove the NVMe over RDMA adapter or the NVMe over TCP adapter.

Working with Datastores in vSphere Storage Environment

9

In the vSphere environment, datastores are logical containers, analogous to file systems, that hide specifics of physical storage and provide a uniform model for storing virtual machine files. You can also use datastores for storing ISO images, virtual machine templates, and floppy images. vCenter Server and ESXi support VMFS, NFS, vSAN, and Virtual Volumes datastores.

vCenter Server and ESXi support the following types of datastores.

Table 9-1. Types of Datastores

Datastore Type	Description
VMFS (version 5 and 6)	Datastores that you deploy on block storage devices use the vSphere Virtual Machine File System (VMFS) format. VMFS is a special high-performance file system format that is optimized for storing virtual machines. See vSphere VMFS Datastore Concepts and Operations.
NFS (version 3 and 4.1)	An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume. The volume is located on a NAS server. The ESXi host mounts the volume as an NFS datastore, and uses it for storage needs. ESXi supports versions 3 and 4.1 of the NFS protocol. See NFS Datastore Concepts and Operations in vSphere Environment
vSAN	vSAN aggregates all local capacity devices available on the hosts into a single datastore shared by all hosts in the vSAN cluster. See the *Administering VMware vSAN* documentation.
vVol	Virtual Volumes datastore represents a storage container in vCenter Server and vSphere Client. See Chapter 14 Working with VMware vSphere Virtual Volumes.

Depending on your storage type, some of the following tasks are available for the datastores.

- Create datastores. You can use the vSphere Client to create certain types of datastores.
- Perform administrative operations on the datastores. Several operations, such as renaming a datastore, are available for all types of datastores. Others apply to specific types of datastores.

- Organize the datastores. For example, you can group them into folders according to business practices. After you group the datastores, you can assign the same permissions and alarms on the datastores in the group at one time.
- Add the datastores to datastore clusters. A datastore cluster is a collection of datastores with shared resources and a shared management interface. When you create the datastore cluster, you can use Storage DRS to manage storage resources. For information about datastore clusters, see the *vSphere Resource Management* documentation.

Read the following topics next:

- vSphere VMFS Datastore Concepts and Operations
- Creating vSphere Datastores
- Basic Administrative Operations for vSphere Datastores
- Managing vSphere VMFS Datastores
- NFS Datastore Concepts and Operations in vSphere Environment
- Using Datastore Browser in vSphere Environment

vSphere VMFS Datastore Concepts and Operations

To store virtual disks, ESXi uses datastores. The datastores are logical containers that hide specifics of physical storage from virtual machines and provide a uniform model for storing the virtual machine files. The datastores that you deploy on block storage devices use the native vSphere Virtual Machine File System (VMFS) format. It is a special high-performance file system format that is optimized for storing virtual machines.

Use the vSphere Client to set up the VMFS datastore in advance on the block-based storage device that your ESXi host discovers.

After creating a datastore, you can perform basic administration operations, such as renaming, unmounting, or removing the datastore. You can also perform more advanced tasks, for example, expand the datastore, manage duplicate datastores, or collect datastore diagnostic information.

The workflow that you use when managing the VMFS datastores generally includes the following actions.

Action	Description
1. Create a vSphere VMFS Datastore	Set up the VMFS datastore on the block-based storage device that your ESXi host discovers.
2. Basic Administrative Operations for vSphere Datastores	Perform basic administration operations, such as renaming, unmounting, or removing the datastore.

Action	Description
3. Managing vSphere VMFS Datastores	Perform more advanced tasks, for example, expand the datastore, manage duplicate datastores, or collect datastore diagnostic information.
4. Using Datastore Browser in vSphere Environment	Use the datastore file browser in your vSphere environment to manage contents of your datastores. You can upload files or folders, download files from the datastore, or move or copy folders or files to a new location. You can also use the datastore browser to inflate the thin virtual disk.

Versions of VMFS Datastores

Several versions of the VMFS file system have been released since its introduction. Currently, ESXi supports VMFS5 and VMFS6.

For all supported VMFS version, ESXi offers complete read and write support. On the supported VMFS datastores, you can create and power on virtual machines.

Table 9-2. Host Access to VMFS Versions

VMFS	ESXi
VMFS6	Read and write
VMFS5	Read and write

The following table compares major characteristics of VMFS5 and VMFS6. For additional information, see *Configuration Maximums* .

Table 9-3. Comparing VMFS5 and VMFS6

Features and Functionalities	VMFS5	VMFS6
Access for ESXi hosts version 6.5 and later	Yes	Yes
Access for ESXi hosts version 6.0 and earlier	Yes	No
Datastores per host	512	512
512n storage devices	Yes	Yes (default)
512e storage devices	Yes. Not supported on local 512e devices.	Yes (default)
4Kn storage devices	No	Yes
Automatic space reclamation	No	Yes
Manual space reclamation through the esxcli command. See Manually Reclaim Accumulated Storage Space on VMFS5.	Yes	Yes

Table 9-3. Comparing VMFS5 and VMFS6 (continued)

Features and Functionalities	VMFS5	VMFS6
Space reclamation from guest OS	Limited	Yes
GPT storage device partitioning	Yes	Yes
MBR storage device partitioning	Yes For a VMFS5 datastore that has been previously upgraded from VMFS3.	No
Storage devices greater than 2 TB for each VMFS extent	Yes	Yes
Support for virtual machines with large capacity virtual disks, or disks greater than 2 TB	Yes	Yes
Support of small files of 1 KB	Yes	Yes
Default use of ATS-only locking mechanisms on storage devices that support ATS. See vSphere VMFS Locking Mechanisms.	Yes	Yes
Block size	Standard 1 MB	Standard 1 MB
Default snapshots	VMFSsparse for virtual disks smaller than 2 TB. SEsparse for virtual disks larger than 2 TB.	SEsparse
Virtual disk emulation type	512n	512n
vMotion	Yes	Yes
Storage vMotion across different datastore types	Yes	Yes
High Availability and Fault Tolerance	Yes	Yes
DRS and Storage DRS	Yes	Yes
RDM	Yes	Yes

When you work with VMFS datastores, consider the following:

- Datastore Extents. A spanned VMFS datastore must use only homogeneous storage devices, either 512n, 512e, or 4Kn. The spanned datastore cannot extend over devices of different formats.
- Block Size. The block size on a VMFS datastore defines the maximum file size and the amount of space a file occupies. VMFS5 and VMFS6 datastores support the block size of 1 MB.
- Storage vMotion. Storage vMotion supports migration across VMFS, vSAN, and Virtual Volumes datastores. vCenter Server performs compatibility checks to validate Storage vMotion across different types of datastores.

- Storage DRS. VMFS5 and VMFS6 can coexist in the same datastore cluster. However, all datastores in the cluster must use homogeneous storage devices. Do not mix devices of different formats within the same datastore cluster.
- Device Partition Formats. Any new VMFS5 or VMFS6 datastore uses GUID partition table (GPT) to format the storage device. The GPT format enables you to create datastores larger than 2 TB. If your VMFS5 datastore has been previously upgraded from VMFS3, it continues to use the master boot record (MBR) partition format, which is characteristic for VMFS3. Conversion to GPT happens only after you expand the datastore to a size larger than 2 TB.

VMFS Datastores as Repositories

ESXi can format SCSI-based storage devices as VMFS datastores. VMFS datastores primarily serve as repositories for virtual machines.

Note Always have only one VMFS datastore for each LUN.

You can store multiple virtual machines on the same VMFS datastore. Each virtual machine, encapsulated in a set of files, occupies a separate single directory. For the operating system inside the virtual machine, VMFS preserves the internal file system semantics, which ensures correct application behavior and data integrity for applications running in virtual machines.

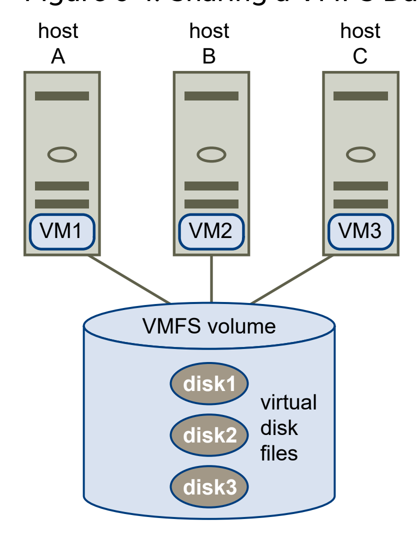
When you run multiple virtual machines, VMFS provides specific locking mechanisms for the virtual machine files. As a result, the virtual machines can operate safely in a SAN environment where multiple ESXi hosts share the same VMFS datastore.

In addition to the virtual machines, the VMFS datastores can store other files, such as the virtual machine templates and ISO images.

Sharing a VMFS Datastore Across Hosts

As a cluster file system, VMFS lets multiple ESXi hosts access the same VMFS datastore concurrently.

Figure 9-1. Sharing a VMFS Datastore Across Hosts



For information on the maximum number of hosts that can connect to a single VMFS datastore, see the *Configuration Maximums* document.

To ensure that multiple hosts do not access the same virtual machine at the same time, VMFS provides on-disk locking. For more details about the ATS-only and ATS+SCSI locking mechanisms that VMFS uses, see vSphere VMFS Locking Mechanisms. You can also learn how to display information about locking mechanisms, how to change from ATS+SCSI to ATS-only, or how to downgrade to ATS+SCSI.

Sharing the VMFS volume across multiple hosts offers several advantages, for example, the following:

- You can use VMware Distributed Resource Scheduling (DRS) and VMware High Availability (HA).

  You can distribute virtual machines across different physical servers. That means you run a mix of virtual machines on each server, so that not all experience high demand in the same area at the same time. If a server fails, you can restart virtual machines on another physical server. If the failure occurs, the on-disk lock for each virtual machine is released. For more information about VMware DRS, see the *vSphere Resource Management* documentation. For information about VMware HA, see the *vSphere Availability* documentation.

- You can use vMotion to migrate running virtual machines from one physical server to another. For information about migrating virtual machines, see the *vCenter Server and Host Management* documentation.

To create a shared datastore, mount the datastore on those ESXi hosts that require the datastore access. See Mount Datastores.

VMFS Metadata Updates

A VMFS datastore holds virtual machine files, directories, symbolic links, RDM descriptor files, and so on. The datastore also maintains a consistent view of all the mapping information for these objects. This mapping information is called metadata.

Metadata is updated each time you perform datastore or virtual machine management operations. Examples of operations requiring metadata updates include the following:

- Creating, growing, or locking a virtual machine file
- Changing attributes of a file
- Powering a virtual machine on or off
- Creating or deleting a VMFS datastore
- Expanding a VMFS datastore
- Creating a template
- Deploying a virtual machine from a template
- Migrating a virtual machine with vMotion

When metadata changes are made in a shared storage environment, VMFS uses special locking mechanisms to protect its data and prevent multiple hosts from concurrently writing to the metadata.

Snapshot Formats on VMFS

When you take a snapshot, the state of the virtual disk is preserved, which prevents the guest operating system from writing to it. A delta or child disk is created. The delta represents the difference between the current state of the VM disk and the state that existed when you took the previous snapshot. On the VMFS datastore, the delta disk is a sparse disk.

Sparse disks use the copy-on-write mechanism, in which the virtual disk contains no data, until the data is copied there by a write operation. This optimization saves storage space.

Depending on the type of your datastore, delta disks use different sparse formats.

Snapshot Formats	VMFS5	VMFS6
VMFSsparse	For virtual disks smaller than 2 TB.	N/A
SEsparse	For virtual disks larger than 2 TB.	For all disks.

VMFSsparse

VMFS5 uses the VMFSsparse format for virtual disks smaller than 2 TB.

VMFSsparse is implemented on top of VMFS. The VMFSsparse layer processes I/Os issued to a snapshot VM. Technically, VMFSsparse is a redo-log that starts empty, immediately after a VM snapshot is taken. The redo-log expands to the size of its base vmdk, when the entire vmdk is rewritten with new data after the VM snapshotting. This redo-log is a file in the VMFS datastore. Upon snapshot creation, the base vmdk attached to the VM is changed to the newly created sparse vmdk.

SEsparse

SEsparse is a default format for all delta disks on the VMFS6 datastores. On VMFS5, SEsparse is used for virtual disks of the size 2 TB and larger.

SEsparse is a format similar to VMFSsparse with some enhancements. This format is space efficient and supports the space reclamation technique. With space reclamation, blocks that the guest OS deletes are marked. The system sends commands to the SEsparse layer in the hypervisor to unmap those blocks. The unmapping helps to reclaim space allocated by SEsparse once the guest operating system has deleted that data. For more information about space reclamation, see Storage Space Reclamation in vSphere.

Snapshot Migration

You can migrate VMs with snapshots across different datastores. The following considerations apply:

- If you migrate a VM with the VMFSsparse snapshot to VMFS6, the snapshot format changes to SEsparse.

- When a VM with a vmdk of the size smaller than 2 TB is migrated to VMFS5, the snapshot format changes to VMFSsparse.
- You cannot mix VMFSsparse redo-logs with SEsparse redo-logs in the same hierarchy.

Upgrading VMFS Datastores

You cannot upgrade a VMFS5 datastore to VMFS6. If you have a VMFS5 datastore in your environment, create a VMFS6 datastore and migrate virtual machines from the VMFS5 datastore to VMFS6.

Creating vSphere Datastores

You use the New Datastore wizard to create your datastores in the vSphere environment. Depending on the type of your storage and storage needs, you can create a VMFS, NFS, or Virtual Volumes datastore.

A vSAN datastore is automatically created when you enable vSAN. For information, see the *Administering VMware vSAN* documentation.

You can also use the New Datastore wizard to manage VMFS datastore copies. See vSphere VMFS Datastore Copies and Datastore Resignaturing.

To learn more about the vSphere datastores, see the following:

- vSphere VMFS Datastore Concepts and Operations
- NFS Datastore Concepts and Operations in vSphere Environment
- Chapter 14 Working with VMware vSphere Virtual Volumes

Create a vSphere VMFS Datastore

VMFS datastores serve as repositories for virtual machines. You can set up VMFS datastores on any SCSI or NVMe based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.

Prerequisites

- Install and configure any adapters that your storage requires.
  - For Fibre Channel, see Setting Up ESXi Fibre Channel Environment.
  - For iSCSI, see Chapter 5 Configuring iSCSI and iSER Adapters and Storage with ESXi.
  - For NVMe, see Chapter 8 About VMware NVMe Storage.
- To discover newly added storage devices, perform a rescan. See Rescan Operations for ESXi Storage.
- Verify that storage devices you are planning to use for your datastores are available. See Viewing Storage Devices Available to an ESXi Host.

Procedure

1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.

2 From the right-click menu, select **Storage > New Datastore**.

3 Select VMFS as the datastore type.

4 Enter the datastore name and if necessary, select the placement location for the datastore.

   The system enforces an 80 character limit for the datastore name.

5 Select the device to use for your datastore.

   Important The device you select must not have any values displayed in the Snapshot Volume column. If a value is present, the device contains a copy of an existing VMFS datastore. For information on managing datastore copies, see vSphere VMFS Datastore Copies and Datastore Resignaturing.

6 Specify the datastore version.

Option	Description
VMFS6	Default format on all hosts that support VMFS6. The ESXi hosts of version 6.0 or earlier cannot recognize the VMFS6 datastore.
VMFS5	VMFS5 datastore supports access by the ESXi hosts of version 6.7 or earlier.

7 Define configuration details for the datastore.

   Note The required minimum size for a VMFS6 datastore is 2 GB.

   a Specify partition configuration.

Option	Description
Use all available partitions	Dedicates the entire disk to a single VMFS datastore. If you select this option, all file systems and data currently stored on this device are destroyed.
Use free space	Deploys a VMFS datastore in the remaining free space of the disk.

   b If the space allocated for the datastore is excessive for your purposes, adjust the capacity values in the Datastore Size field.

      By default, the entire free space on the storage device is allocated.

   c For VMFS6, specify the block size and define space reclamation parameters. See Space Reclamation on vSphere VMFS Datastores.

8 In the Ready to Complete page, review the datastore configuration information and click **Finish**.

Results

The datastore on the SCSI or NVMe based storage device is created. It is available to all hosts that have access to the device.

What to do next

After you create the VMFS datastore, you can perform the following tasks:

- Change the capacity of the datastore. See Increase Capacity of vSphere VMFS Datastore .
- Edit space reclamation settings. See Configure Fixed Reclamation for the Existing VMFS6 Datastore.
- Enable shared vmdk support. See Configure Settings for Clustered Virtual Disks on the vSphere VMFS6 Datastore.

Create an NFS Datastore in vSphere Environment

You can use the **New Datastore** wizard to mount an NFS volume.

Prerequisites

- Set up NFS storage environment.
- If you plan to use Kerberos authentication with the NFS 4.1 datastore, make sure to configure the ESXi hosts for Kerberos authentication.
- If you plan to isolate the NFS 3 traffic to a specific VMkernel adapter that you connect to the datastore, create and configure the VMkernel adapter on default TCP/IP stack. For more information, see Configure VMkernel Binding for NFS 3 Datastores.

Procedure

1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.

2 From the right-click menu, select **Storage > New Datastore**.

3 Select NFS as the datastore type and specify an NFS version.

   - NFS 3
   - NFS 4.1

   Important If multiple hosts access the same datastore, you must use the same protocol on all hosts.

4 Enter the datastore parameters.

Option	Description
Datastore name	The system enforces an 80 character limit for the datastore name.
Folder	The mount point folder name
Server	The server name or IP address. You can use IPv6 or IPv4 formats. With NFS 4.1, you can add multiple IP addresses or server names if the NFS server supports trunking. The ESXi host uses these values to achieve multipathing to the NFS server mount point.

5 Select **Mount NFS read only** if the volume is exported as read-only by the NFS server.

6 To connect the NFS 3 datastore to a specific VMkernel adapter, select **Bind to vmknic**. For more information, see Configure VMkernel Binding for NFS 3 Datastores.

Note Only NFS 3 datastores support connections to VMkernel adapters.

7 To use Kerberos security with NFS 4.1, enable Kerberos and select an appropriate Kerberos model.

Option	Description
Use Kerberos for authentication only (krb5)	Supports identity verification
Use Kerberos for authentication and data integrity (krb5i)	In addition to identity verification, provides data integrity services. These services help to protect the NFS traffic from tampering by checking data packets for any potential modifications.

If you do not enable Kerberos, the datastore uses the default AUTH_SYS security.

8 If you create a datastore at the data center or cluster level, select hosts that mount the datastore.

9 Review the configuration options and click **Finish**.

Create a Virtual Volumes Datastore in vSphere Environment

You use the **New Datastore** wizard to create a Virtual Volumes datastore.

For more information about the Virtual Volumes datastore, see Virtual Volumes Datastores.

Procedure

1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.

2 From the right-click menu, select **Storage > New Datastore**.

3 Select **vVol** as the datastore type.

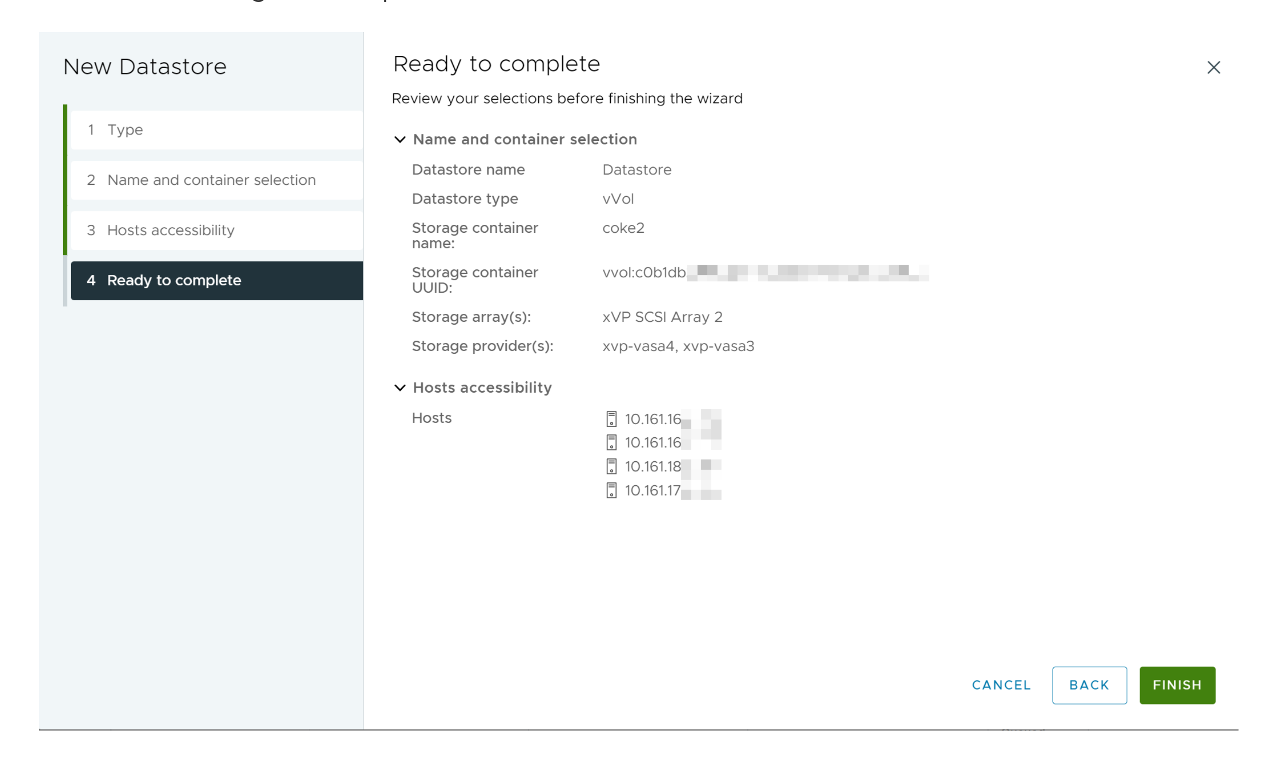
4 Enter the datastore name and select a backing storage container from the list of storage containers.

   Make sure to use the name that does not duplicate another datastore name in your data center environment.

   If you mount the same Virtual Volumes datastore to several hosts, the name of the datastore must be consistent across all hosts.

5 Select the hosts that require access to the datastore.

6 Review the configuration options and click **Finish**.



What to do next

After you create the Virtual Volumes datastore, you can perform the following operations:

- Rename the datastore, browse datastore files, unmount the datastore, and so on. See Basic Administrative Operations for vSphere Datastores.

  Note You cannot add the Virtual Volumes datastore to a datastore cluster.

- Provision virtual machines on the Virtual Volumes datastore. See Provision VMs on vSphere Virtual Volumes Datastores and the *vSphere Virtual Machine Administration* documentation.
- Review and manage protocol endpoints. See Review and Manage Protocol Endpoints in vSphere.
- Manage or upgrade storage providers. See Manage Storage Providers for vSphere Virtual Volumes.

Basic Administrative Operations for vSphere Datastores

After creating datastores, you can perform several administrative operations on the datastores, such as renaming, unmounting, or removing the datastore. Certain operations, for example, renaming, are available for all types of datastores. Others, such as deleting the datastore, apply only to VMFS datastores.

Change Datastore Name

Use the vSphere Client to change the name of an existing datastore. You can rename the datastore that has virtual machines running on it without any negative impact.

Note If the host is managed by vCenter Server, you cannot rename the datastore by directly accessing the host from the VMware Host Client. You must rename the datastore from vCenter Server.

Procedure

1 In the vSphere Client, navigate to the datastore.

2 Right-click the datastore to rename, and select **Rename**.

3 Enter a new datastore name.

   The system enforces an 80 character limit for the datastore name.

Results

The new name appears on all hosts that have access to the datastore.

Unmount Datastores

When you unmount a datastore, it remains intact, but can no longer be seen from the hosts that you specify. The datastore continues to appear on other hosts, where it remains mounted.

Do not perform any configuration operations that might result in I/O to the datastore while the unmounting is in progress.

Note Make sure that the datastore is not used by vSphere HA Heartbeating. vSphere HA Heartbeating does not prevent you from unmounting the datastore. However, if the datastore is used for heartbeating, unmounting it might cause the host to fail and restart any active virtual machine.

Prerequisites

When appropriate, before unmounting datastores, make sure that the following prerequisites are met:

- No virtual machines reside on the datastore.
- Storage DRS does not manage the datastore.

- Storage I/O Control is deactivated for this datastore.

Procedure

1. In the vSphere Client, navigate to the datastore.
2. Right-click the datastore and select **Unmount Datastore**.
3. If the datastore is shared, select the hosts from which to unmount the datastore.
4. Confirm that you want to unmount the datastore.

Results

After you unmount a VMFS datastore from all hosts, the datastore is marked as inactive. If you unmount an NFS or a Virtual Volumes datastore from all hosts, the datastore disappears from the inventory. You can mount the unmounted VMFS datastore. To mount the NFS or Virtual Volumes datastore that has been removed from the inventory, use the New Datastore wizard.

What to do next

If you unmounted the VMFS datastore as a part of a storage removal procedure, you can now detach the storage device that is backing the datastore. See Detach Storage Devices.

Mount Datastores

You can mount a datastore you previously unmounted. You can also mount a datastore on additional hosts, so that it becomes a shared datastore.

A VMFS datastore that has been unmounted from all hosts remains in inventory, but is marked as inaccessible. See Unmount Datastores.

You can use this task to mount the VMFS datastore to a specified host or multiple hosts.

If you have unmounted an NFS or a Virtual Volumes datastore from all hosts, the datastore disappears from the inventory. To mount the NFS or Virtual Volumes datastore that has been removed from the inventory, use the New Datastore wizard.

A datastore of any type that is unmounted from some hosts while being mounted on others, is shown as active in the inventory.

Procedure

1. In the vSphere Client, navigate to the datastore.
2. Right-click the datastore to mount and select one of the following options:
   - **Mount Datastore**
   - **Mount Datastore on Additional Hosts**

     You can also select the VMkernel binding option for the required datastore. For more information, see step g under Configure VMkernel Binding for NFS 3 Datastores.

   Whether you see one or another option depends on the type of datastore you use.

3 Select the hosts that should access the datastore and click **OK**.

4 To list all hosts that share the datastore, navigate to the datastore, and click the **Hosts** tab.

Remove VMFS Datastores

You can delete any type of VMFS datastore, including copies that you have mounted without resignaturing. When you delete a datastore, it is destroyed and disappears from all hosts that have access to the datastore.

Note The delete operation for the datastore permanently deletes all files associated with virtual machines on the datastore. Although you can delete the datastore without unmounting, it is preferable that you unmount the datastore first.

Prerequisites

- Remove or migrate all virtual machines from the datastore.
- Unmount the datastore from all hosts.
- Disable Storage DRS for the datastore. For more information, see Activate and Deactivate Storage DRS.
- Disable Storage I/O Control for the datastore. For more information, see Enable Storage I/O Control.
- Make sure that the datastore is not used for vSphere HA heartbeating. For more information, see Configure Heartbeat Datastores.

Procedure

1 In the vSphere Client, navigate to the datastore.

2 Right-click the datastore to remove, and select **Delete Datastore**.

3 Confirm that you want to remove the datastore.

Managing vSphere VMFS Datastores

After you create a VMFS datastore in your vSphere environment, you can perform such tasks as expanding the datastore, managing duplicate datastores, or collecting datastore diagnostic information.

For general information about VMFS datastores, see vSphere VMFS Datastore Concepts and Operations.

To create a VMFS datastore and perform basic operations, see these topics:

- Create a vSphere VMFS Datastore
- Basic Administrative Operations for vSphere Datastores

vSphere VMFS Locking Mechanisms

In a shared storage environment, when multiple ESXi hosts access the same VMFS datastore, locking mechanisms are used. The locking mechanisms prevent multiple hosts from concurrently writing to the metadata and ensure that no data corruption occurs. VMFS supports ATS-only and ATS+SCSI locking. Learn how to display information about locking mechanisms, how to change from ATS+SCSI to ATS-only, or how to downgrade to ATS+SCSI.

To learn more about VMFS as a cluster file system, see vSphere VMFS Datastore Concepts and Operations.

Depending on its configuration and the type of underlying storage, a VMFS datastore can use different types of locking mechanisms. It can exclusively use the atomic test and set locking mechanism (ATS-only), or use a combination of ATS and SCSI reservations (ATS+SCSI).

ATS-Only Mechanism

For storage devices that support T10 standard-based VAAI specifications, VMFS provides ATS locking, also called hardware assisted locking. The ATS algorithm supports discrete locking per disk sector. All newly formatted VMFS5 and VMFS6 datastores use the ATS-only mechanism if the underlying storage supports it, and never use SCSI reservations.

When you create a multi-extent datastore where ATS is used, vCenter Server filters out non-ATS devices. This filtering allows you to use only those devices that support the ATS primitive.

In certain cases, you might need to change the default locking mechanism for a VMFS5 or VMFS6 datastore. For information, see Change Locking Mechanism to ATS+SCSI.

Note If you are running a VMware vSAN environment or have ATS-only VMFS volumes, do not deactivate ATS. Deactivating ATS might result in an outage as no locking mechanism is available. For more information, see a VMware Knowledge Base article.

ATS+SCSI Mechanism

A VMFS datastore that supports the ATS+SCSI mechanism is configured to use ATS and attempts to use it when possible. If ATS fails, the VMFS datastore reverts to SCSI reservations. In contrast with the ATS locking, the SCSI reservations lock an entire storage device while an operation that requires metadata protection is performed. After the operation completes, VMFS releases the reservation, and other operations can continue.

Datastores that use the ATS+SCSI mechanism include VMFS5 datastores that were upgraded from VMFS3. In addition, new VMFS5 or VMFS6 datastores on storage devices that do not support ATS use the ATS+SCSI mechanism.

If the VMFS datastore reverts to SCSI reservations, you might notice performance degradation caused by excessive SCSI reservations.

Display VMFS Locking Information

Use the `esxcli` command to obtain information about the locking mechanism that a VMFS datastore uses.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To display information related to VMFS locking mechanisms, run the following command:

  `esxcli storage vmfs lockmode list`

Results

The table lists items that the output of the command might include.

Table 9-4. VMFS Locking Information

Fields	Values	Descriptions
Locking Modes		Indicates the locking configuration of the datastore.
	`ATS`	The datastore is configured to use the ATS-only locking mode.
	`ATS+SCSI`	The datastore is configured to use the ATS mode. If ATS fails or is not supported, the datastore can revert to SCSI.
	`ATS upgrade pending`	The datastore is in the process of an online upgrade to the ATS-only mode.
	`ATS downgrade pending`	The datastore is in the process of an online downgrade to the ATS+SCSI mode.
ATS Compatible		Indicates whether the datastore can be or cannot be configured for the ATS-only mode.
ATS Upgrade Modes		Indicates the type of upgrade that the datastore supports.
	`None`	The datastore is not ATS-only compatible.
	`Online`	The datastore can be used during its upgrade to the ATS-only mode.
	`Offline`	The datastore cannot be used during its upgrade to the ATS-only mode.
ATS Incompatibility Reason		If the datastore is not compatible with ATS-only, the item indicates the reason for the incompatibility.

ATS-Only Locking Mechanism for VMFS

If your VMFS datastore uses the ATS+SCSI locking mechanism, you can change to ATS-only locking.

Typically, VMFS5 datastores that were previously upgraded from VMFS3 continue using the ATS+SCSI locking mechanism. If the datastores are deployed on ATS-enabled hardware, they are eligible for an upgrade to ATS-only locking. Depending on your vSphere environment, you can use one of the following upgrade modes:

- The online upgrade to the ATS-only mechanism is available for most single-extent VMFS5 datastores. While you perform the online upgrade on one of the hosts, other hosts can continue using the datastore.
- The offline upgrade to ATS-only must be used for VMFS5 datastores that span multiple physical extents. Datastores composed of multiple extents are not eligible for the online upgrade. These datastores require that no hosts actively use the datastores at the time of the upgrade request.

Prepare for an Upgrade to ATS-Only Locking

You must perform several steps to prepare your environment for an online or offline upgrade to ATS-only locking.

Procedure

1 Upgrade all hosts that access the VMFS5 datastore to the newest version of vSphere.

2 Determine whether the datastore is eligible for an upgrade of its current locking mechanism by running the `esxcli storage vmfs lockmode list` command.

   The following sample output indicates that the datastore is eligible for an upgrade. It also shows the current locking mechanism and the upgrade mode available for the datastore.

```
Locking Mode  ATS Compatible  ATS Upgrade Modes
------------  --------------  -----------------
ATS+SCSI           true       Online or Offline
```

3 Depending on the upgrade mode available for the datastore, perform one of the following actions:

Upgrade Mode	Action
Online	Verify that all hosts have consistent storage connectivity to the VMFS datastore.
Offline	Verify that no hosts are actively using the datastore.

Upgrade Locking Mechanism to the ATS-Only Type

If a VMFS datastore is ATS-only compatible, you can upgrade its locking mechanism from ATS+SCSI to ATS-only.

Most datastores that do not span multiple extents are eligible for an online upgrade. While you perform the online upgrade on one of the ESXi hosts, other hosts can continue using the datastore. The online upgrade completes only after all hosts have closed the datastore.

Prerequisites

If you plan to complete the upgrade of the locking mechanism by putting the datastore into maintenance mode, deactivate Storage DRS. This prerequisite applies only to an online upgrade.

Procedure

1 Perform an upgrade of the locking mechanism by running the following command:

   `esxcli storage vmfs lockmode set -a|--ats -l|--volume-label=` *VMFS label* `-u|--volume-uuid=` *VMFS UUID*.

2 For an online upgrade, perform additional steps.

   a Close the datastore on all hosts that have access to the datastore, so that the hosts can recognize the change.

      You can use one of the following methods:

      - Unmount and mount the datastore.
      - Put the datastore into maintenance mode and exit maintenance mode.

   b Verify that the Locking Mode status for the datastore changed to ATS-only by running:

      ```
      esxcli storage vmfs lockmode list
      ```

   c If the Locking Mode displays any other status, for example ATS UPGRADE PENDING, check which host has not yet processed the upgrade by running:

      ```
      esxcli storage vmfs host list
      ```

Change Locking Mechanism to ATS+SCSI

When you create a VMFS5 datastore on a device that supports the atomic test and set (ATS) locking, the datastore uses the ATS-only locking mechanism. In certain circumstances, you might need to downgrade the ATS-only locking to ATS+SCSI.

You might need to switch to the ATS+SCSI locking mechanism when, for example, your storage device is downgraded. Or when firmware updates fail and the device no longer supports ATS.

The downgrade process is similar to the ATS-only upgrade. As with the upgrade, depending on your storage configuration, you can perform the downgrade in online or offline mode.

Note If you are running a VMware vSAN environment or have ATS-only VMFS volumes, do not deactivate ATS. Deactivating ATS might result in an outage as no locking mechanism is available. For more information, see a VMware Knowledge Base article.

Procedure

1 Change the locking mechanism to ATS+SCSI by running the following command:

   `esxcli storage vmfs lockmode set -s|--scsi -l|--volume-label=` *VMFS label* `-u|--volume-uuid=` *VMFS UUID*.

2 For an online mode, close the datastore on all hosts that have access to the datastore, so that the hosts can recognize the change.

vSphere VMFS Datastore Copies and Datastore Resignaturing

When a storage device contains a VMFS datastore copy, ESXi detects the copy. You can mount the datastore copy with the original datastore UUID or change the UUID. The process of changing the UUID is called datastore resignaturing.

Each VMFS datastore created on a storage device has a unique signature, also called UUID, that is stored in the file system superblock. When the storage device is replicated or its snapshot is taken on the array side, the resulting device copy is identical, byte-for-byte, with the original device. For example, if the original storage device contains a VMFS datastore with UUIDX, the copy appears to contain a datastore copy with the same UUIDX.

In addition to LUN snapshots and replications, certain device operations, such as LUN ID changes, might produce a copy of the original datastore.

After the ESXi host detects the VMFS datastore copy, you can mount the datastore copy with its original UUID or change the UUID using the datastore resignaturing.

Whether you select resignaturing or mounting without resignaturing depends on how the LUNs are masked in the storage environment. If your hosts can see both copies of the LUN, then resignaturing is the optimal method.

Keeping Existing Datastore Signature

If you do not need to resignature a VMFS datastore copy, you can mount it without changing its signature.

You can keep the signature if, for example, you maintain synchronized copies of virtual machines at a secondary site as part of a disaster recovery plan. In the event of a disaster at the primary site, you mount the datastore copy and power on the virtual machines at the secondary site.

Resignaturing a VMFS Datastore Copy

Use datastore resignaturing if you want to retain the data stored on the VMFS datastore copy.

When resignaturing a VMFS copy, ESXi assigns a new signature (UUID) to the copy, and mounts the copy as a datastore distinct from the original. All references to the original signature in virtual machine configuration files are updated.

When you perform datastore resignaturing, consider the following points:

- Datastore resignaturing is irreversible.

- After resignaturing, the storage device replica that contained the VMFS copy is no longer treated as a replica.
- A spanned datastore can be resignatured only if all its extents are online.
- The resignaturing process is fault tolerant. If the process is interrupted, you can resume it later.
- You can mount the new VMFS datastore without a risk of its UUID conflicting with UUIDs of any other datastore from the hierarchy of device snapshots.

Mount a VMFS Datastore Copy

Use datastore resignaturing if you want to retain the data stored on the VMFS datastore copy. If you do not need to resignature the VMFS datastore copy, you can mount it without changing its signature.

Prerequisites

- Perform a storage rescan on your host to update the view of storage devices presented to the host.
- Unmount the original VMFS datastore that has the same UUID as the copy you plan to mount. You can mount the VMFS datastore copy only if it does not collide with the original VMFS datastore.

Procedure

1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.

2 From the right-click menu, select **Storage > New Datastore**.

3 Select VMFS as the datastore type.

4 Enter the datastore name and if necessary, select the placement location for the datastore.

5 From the list of storage devices, select the device that has a specific value displayed in the Snapshot Volume column.

   The value present in the Snapshot Volume column indicates that the device is a copy that contains a copy of an existing VMFS datastore.

6 Mount the datastore.

Option	Description
Mount with resignaturing	Under **Mount Options**, select **Assign a New Signature** and click **Next** .
Mount without resignaturing	Under Mount Options, select **Keep Existing Signature**.

7 Review the datastore configuration information and click **Finish**.

8 To make sure that the storage is presented uniformly to all hosts in the cluster, perform a rescan.

   See Perform Storage Rescan.

Increase Capacity of vSphere VMFS Datastore

You can increase the capacity of a VMFS datastore on the ESXi host. Additional capacity might be required when you add virtual machines to the datastore, or when the virtual machines running on the datastore require more space.

If a shared datastore has powered on virtual machines and becomes 100% full, you can increase the datastore capacity. You can perform this action only from the host where the powered on virtual machines are registered.

Depending on your storage configuration, you can use one of the following methods to increase the datastore capacity. You do not need to power off virtual machines when using either method of increasing the datastore capacity.

Expand an Existing Datastore

Increase the size of an expandable datastore. The datastore is considered expandable when the backing storage device has free space immediately after the datastore extent.

Add an Extent

Increase the capacity of an existing VMFS datastore by adding new storage devices to the datastore. The datastore can span over multiple storage devices, yet appear as a single volume.

The spanned VMFS datastore can use any or all its extents at any time. It does not need to fill up a particular extent before using the next one.

Note Datastores that support only the hardware assisted locking, also called the atomic test and set (ATS) mechanism, cannot span over non-ATS devices. For more information, see vSphere VMFS Locking Mechanisms.

Prerequisites

You can increase the datastore capacity if the host storage meets one of the following conditions:

- The backing device for the existing datastore has enough free space.
- You added new storage devices to the host.

Procedure

1 In the vSphere Client, navigate to the datastore.

2 Select **Increase Datastore Capacity** from the datastore right-click menu.

3 Select a device from the list of storage devices.

Your selection depends on whether an expandable storage device is available.

Option	Description
To expand an existing datastore extent	Select the device for which the Expandable column reads YES.
To add an extent	Select the device for which the Expandable column reads NO.

4 Review the **Partition Layout** to see the available configurations.

5 Select a configuration option from the bottom panel.

Depending on the current layout of the disk and on your previous selections, the menu items you see might vary.

Menu Item	Description
Use free space to expand the datastore	Expands an existing extent to a required capacity.
Use free space	Deploys an extent in the remaining free space of the disk. This menu item is available only when you are adding an extent.
Use all available partitions	Dedicates the entire disk to a single extent. This menu item is available only when you are adding an extent and when the disk you are formatting is not blank. The disk is reformatted, and the datastores and any data that it contains are erased.

6 Set the capacity for the extent.

The minimum extent size is 1.3 GB. By default, the entire free space on the storage device is available.

7 Click **Next**.

8 Review the proposed layout and the new configuration of your datastore, and click **Finish**.

Note It is recommended to perform a cluster level rescan at vCenter Server after you perform any form of expansion done at the host level. If you do not perform a cluster level rescan from vCenter Server, it may not detect any change. This causes a disparity in the current space/usage for the datastore.

If the device where you increased the capacity is exposed to multiple hosts in multiple clusters, then the cluster level rescan must be triggered on all the clusters where the device is visible.

Configure Settings for Clustered Virtual Disks on the vSphere VMFS6 Datastore

If you plan to use a virtual disk in Windows Server Failover Clustering (WSFC) configurations, your VMFS6 datastore must support clustered virtual disks. Use the vSphere Client to configure the clustered disk support.

For information on the use of clustered virtual disks in VM clusters, see the *Setup for Windows Server Failover Clustering* documentation.

Prerequisites

Follow these guidelines when you use a datastore for clustered virtual disks:

- The storage array supports the ATS, Write Exclusive – All Registrant (WEAR) SCSI-3 reservation type. For this type of configurations, ESXi supports only Fibre Channel arrays.
- Storage devices that back the datastore are claimed by the NMP. ESXi does not support third-party plug-ins (MPPs) in the clustered virtual disk configurations.
- The datastore is of VMFS6 type. You cannot expand or span the datastore.
- The VMFS6 datastore is deployed on the storage device with the attribute `Clustered VMDK Supported` set to `Yes`.
- Virtual disks you use for clustering are in the Thick Provision Eager Zeroed format.

Procedure

1. In the vSphere Client, navigate to the datastore.
2. Click the **Configure** tab, and click **General**.
3. Under **Datastore Capabilities**, click one of the following options next to the **Clustered VMDK** item.

Option	Description
Enable	To activate support for clustered virtual disks on the datastore. After you activate the support, you can place the clustered virtual disks on this VMFS datastore.
Disable	To deactivate the support. Before deactivating, make sure to power off all virtual machines with the clustered virtual disks.

4. Confirm your configuration.

Manage Filters for LUNs and vSphere VMFS Datastores

When you perform VMFS datastore management operations, vCenter Server uses default storage protection filters. The filters help you to avoid storage corruption by retrieving only the storage devices that can be used for a particular operation. Unsuitable devices are not displayed for selection. You can turn off the filters to view all devices.

These filters are available by default.

Filter Name	Description
config.vpxd.filter.vmfsFilter (VMFS Filter)	Filters out storage devices, or LUNs, that are already used by a VMFS datastore on any host managed by vCenter Server. The LUNs do not show up as candidates to be formatted with another VMFS datastore or to be used as an RDM.
config.vpxd.filter.rdmFilter (RDM Filter)	Filters out LUNs that are already referenced by an RDM on any host managed by vCenter Server. The LUNs do not show up as candidates to be formatted with VMFS or to be used by a different RDM. For your virtual machines to access the same LUN, the virtual machines must share the same RDM mapping file. For information about this type of configuration, see the *vSphere Resource Management* documentation.
config.vpxd.filter.sameHostsAndTransportsFilter (Same Hosts and Transports Filter)	Filters out LUNs ineligible for use as VMFS datastore extents because of host or storage type incompatibility. Prevents you from adding the following LUNs as extents: ■ LUNs not exposed to all hosts that share the original VMFS datastore. ■ LUNs that use a storage type different from the one the original VMFS datastore uses. For example, you cannot add a Fibre Channel extent to a VMFS datastore on a local storage device.
config.vpxd.filter.hostRescanFilter (Host Rescan Filter)	Automatically rescans and updates VMFS datastores after you perform datastore management operations. The filter helps provide a consistent view of all VMFS datastores on all hosts managed by vCenter Server. **Note** If you present a new LUN to a host or a cluster, the hosts automatically perform a rescan no matter whether you have the Host Rescan Filter on or off.

Prerequisites

Before you change the device filters, consult with the VMware support team.

Procedure

1 Browse to the vCenter Server instance.

2 Click the **Configure** tab.

3 Under **Settings**, click **Advanced Settings**, and click **EDIT SETTINGS**.

4 Specify the filter to turn off.

In the **Name** and **Value** text boxes at the bottom of the page, enter appropriate information.

Name	Value
config.vpxd.filter.vmfsFilter	False
config.vpxd.filter.rdmFilter	False
config.vpxd.filter.sameHostsAndTransportsFilter	False
config.vpxd.filter.hostRescanFilter	False **Note** If you turn off this filter, your hosts continue to perform a rescan each time you present a new LUN to a host or a cluster.

5 Click **ADD**, and click **SAVE** to save your changes.

You are not required to restart the vCenter Server system.

Set Up Dynamic Disk Mirroring for Virtual Machines on vSphere VMFS

Typically, in the ESXi environment, you cannot use LUN manager software on virtual machines to mirror virtual disks. However, if your Microsoft Windows virtual machines support dynamic disks, you can mirror virtual disks across two SAN LUNs. Mirroring helps you to protect the virtual machines from an unplanned storage device loss.

Prerequisites

- Use a Windows virtual machine that supports dynamic disks.
- Required privilege: **Virtual machine. Configuration. Settings**

Procedure

1 Create a virtual machine with two virtual disks.

Place the disks on different datastores.

2 Log in to your virtual machine and configure the disks as dynamic mirrored disks.

See Microsoft documentation.

3 After the disks synchronize, power off the virtual machine.

4 Change virtual machine settings to enable the dynamic disk mirroring.

a Right-click the virtual machine and select **Edit Settings**.

b Click the **VM Options** tab and expand the **Advanced** menu.

c Click **Edit Configuration** next to Configuration Parameters.

d Click **Add Configuration Params** and add the following parameters:

Replace *#* with the boot device SCSI bus controller ID number. By default, it is set to 0.

Name	Value
`scsi#.returnNoConnectDuringAPD`	True
`scsi#.returnBusyOnNoConnectStatus`	False

e For ESXi version 6.7 or later, include an additional parameter for each virtual disk participating in the software RAID-1 configuration.

The parameter prevents guest OS I/O failures when a storage device fails.

Replace *#* with the boot device SCSI bus controller ID number. By default, it is set to 0.

Name	Value
`scsi#:1.passthruTransientErrors`	True
`scsi#:2.passthruTransientErrors`	True

f Click **OK**.

Setting Up ESXi Core Dump on a VMFS Datastore

During a host failure, ESXi must be able to save diagnostic information to a preconfigured location for diagnostic and technical support purposes.

Typically, a partition to collect diagnostic information, also called a core dump, is created on a local storage device during ESXi installation. You can also configure an ESXi Dump Collector and keep core dumps on a network server for use during debugging. For information on setting up the ESXi Dump Collector, see Configure an ESXi Dump Collector in the *VMware ESXi Installation and Setup* documentation.

Another option is to use a file on a VMFS datastore to collect the diagnostic information.

Set Up a File as Core Dump Location

If the size of your available core dump partition is insufficient, you can configure ESXi to use a file on a VMFS datastore for diagnostic information.

Note VMFS datastores on software iSCSI do not support core dump files.

Prerequisites

- Make sure that the amount of available space in the datastore for a core dump file is twice as large as the recommended file size. For more information, see the Knowledge Base article 2058416.
- Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Create a VMFS datastore core dump file by running the following command:

```
esxcli system coredump file add
```

The command takes the following options, but they are not required and can be omitted:

Option	Description
--datastore | -d *datastore_UUID or datastore_name*	Specify the datastore for the dump file. If not provided, the system selects a datastore of sufficient size.
--file | -f *file_name*	Specify the filename of the dump file. If not provided, the system creates a unique name for the file.
--size |-s *file_size_MB*	Set the size in MB of the dump file. If not provided, the system creates a file of the size appropriate for the memory installed in the host.

2 Verify that the file has been created:

esxcli system coredump file list

You can see the output similar to the following:

```
Path                                              Active Configured Size
------------------------------------------------- ------ ---------- ---------
/vmfs/volumes/52b021c3-.../vmkdump/test.dumpfile false  false      104857600
```

3 Activate the core dump file for the host:

esxcli system coredump file set

The command takes the following options:

Option	Description
--enable |-e	Activate or deactivate the dump file. This option cannot be specified when unconfiguring the dump file.
--path | -p	The path of the core dump file to use. The file must be pre-allocated.
--smart | -s	This flag can be used only with **--enable | -e=true**. It causes the file to be selected using the smart selection algorithm. For example, **esxcli system coredump file set --smart --enable true**
--unconfigure | -u	Unconfigure the current VMFS dump file.

4 Verify that the core dump file is active and configured:

esxcli system coredump file list

The output similar to the following indicates that the core dump file is active and configured:

```
Path                                              Active Configured Size
------------------------------------------------- ------ ---------- ---------
/vmfs/volumes/52b021c3-.../vmkdump/test.dumpfile True   True       104857600
```

What to do next

For information about other commands you can use to manage the core dump files, see the *ESXCLI Reference* documentation.

Deactivate and Delete a Core Dump File

Deactivate a configured core dump file and, if needed, remove it from the VMFS datastore.

You can temporarily deactivate the core dump file. If you do not plan to use the deactivated file, you can remove it from the VMFS datastore. To remove the file that has not been deactivated, you can use the `esxcli system coredump file remove` command with the `--force | -F` parameter.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 List the core dump files:

   `esxcli system coredump file list`

2 Deactivate the core dump file by running the following command:

   `esxcli system coredump file set --unconfigure | -u`

3 Remove the file from the VMFS datastore:

   `esxcli system coredump file remove --file | -f` ***file_name***

   The command takes the following options:

Option	Description
`--file | -f`	Enter the name of the dump file to be removed. If you do not enter the name, the command removes the default configured core dump file.
`--force | -F`	Deactivate and unconfigure the dump file being removed. This option is required if the file has not been previously deactivated and is active.

Results

The core dump file is removed from the VMFS datastore.

Checking Metadata Consistency with vSphere On-disk Metadata Analyzer

Use the vSphere On-disk Metadata Analyzer (VOMA) to check metadata consistency and identify and fix incidents of metadata corruption on the VMFS datastores or logical volumes that back the VMFS datastores.

Problem

You can check metadata consistency when you experience problems with a VMFS datastore or a virtual flash resource. For example, perform a metadata check if one of the following occurs:

- You experience storage outages.
- After you rebuild RAID or perform a disk replacement.
- You see metadata errors in the `vmkernel.log` file similar to the following:

```
cpu11:268057)WARNING: HBX: 599: Volume 50fd60a3-3aae1ae2-3347-0017a4770402
("<Datastore_name>") may be damaged on disk. Corrupt heartbeat detected at offset 3305472:
[HB state 0 offset 6052837899185946624 gen 15439450 stampUS 5 $
```

- You are unable to access files on a VMFS.
- You see corruption being reported for a datastore in events tabs of vCenter Server.

Solution

To check metadata consistency, run VOMA from the CLI of an ESXi host. VOMA can be used to check and fix minor inconsistency issues for a VMFS datastore or logical volumes that back the VMFS datastore.

VOMA can check and fix the following items.

Table 9-5. VOMA Functions

VOMA Functions	Description
Metadata check and fix	Examples of metadata check and fix include, but are not limited to, the following: ■ Validation of VMFS volume header for basic metadata consistency. ■ Checking consistency of VMFS resource files (system file). ■ Checking the pathname and connectivity of all files.
Affinity metadata check and fix	To enable the affinity check for VMFS6, use the `-a|--affinityChk` option. Several examples of affinity metadata check and fix include the following: ■ Affinity flags in resource types and FS3_ResFileMetadata. ■ Validation of affinity flags in SFB RC meta (FS3_ResourceClusterMDVMFS6). ■ Validation of all entries in the affinityInfo entries in rcMeta of RC, including the overflow key, to make sure that no invalid entries exist. Checking for missing entries.

Table 9-5. VOMA Functions (continued)

VOMA Functions	Description
Directory validation	VOMA can detect and correct the following errors: ■ Directory hash block corruption. ■ Alloc map corruption. ■ Link blocks corruptions. ■ Directory entry block corruptions. Based on the nature of the corruption, VOMA can either fix only the corrupted entries or fully reconstruct the hash block, alloc map blocks and link blocks.
Lost and found files	During a filesystem check, VOMA can find files that are not referenced anywhere in the filesystem. These orphaned files are valid and complete, but do not have a name or directory entry on the system. If VOMA encounters orphaned files during scanning, it creates a directory named lost+found at the root of the volume to store the orphaned files. The names of the files use the `File`*`sequence-number`* format.

Command options that the VOMA tool takes include the following.

Table 9-6. VOMA Command Options

Command Option	Description	
`-m|--module`	The modules to run include the following:	
	`vmfs`	If you do not specify the name of the module, this option is used by default. You can check the VMFS file systems and the file systems that back virtual flash resources. If you specify this module, minimal checks are performed for LVM as well.
	`lvm`	Check logical volumes that back the VMFS datastores.
	`ptbl`	Check and validate VMFS partitions, such as MBR or GPT. If no partition exists, determine whether partitions should exist.
`-f|--func`	Functions to be performed include the following:	
	`query`	List functions supported by module.
	`check`	Check for errors.
	`fix`	Check and fix errors.
	`dump`	Collect metadata dump.
`-a|--affinityChk`	Include affinity-related check and fix for VMFS6.	

Table 9-6. VOMA Command Options (continued)

Command Option	Description
`-d|--device`	Indicate the device or disk to inspect. Make sure to provide the absolute path to the device partition backing the VMFS datastore. If the datastore spans multiple devices, provide the UUID of the head extent. For example, `voma -m vmfs -f check -d /vmfs/devices/disks/naa.xxxx:x` If you use the `-x|--extractDump` command, enter multiple device paths, with a partition qualifier, separated with a comma. The number of device paths you enter equals the number of spanned devices.
`-b|--blockSize`	Indicate the disk block size.
`-s|--logfile`	Specify the path to log file to output the results.
`-x|--extractDump`	Extract the collected dump using VOMA.
`-D|--dumpfile`	Indicate the dump file to save the collected metadata dump.
`-v|--version`	Display the version of VOMA.
`-h|--help`	Display the help message for the VOMA command.
`-Y`	Indicate that you run VOMA without using PE tables for address resolution.
`-Z| --file`	Indicate that you run VOMA on extracted device files.

Example

Collect metadata dump from a spanned volume:

voma -m vmfs -f dump -d *head_extent* -D *dump_filename*

Extract collected dump back to devices of a spanned volume:

voma -x *dump_filename* -d *head_extent,extent_2,extent_3...extent_n*

Use VOMA to Check Metadata Consistency

The task demonstrates how to use VOMA to check VMFS metadata consistency. VOMA can be used to check and fix minor inconsistency issues for a VMFS datastore or a virtual flash resource. Run VOMA from the CLI of an ESXi host.

Prerequisites

Power off any virtual machines that are running or migrate them to a different datastore.

Procedure

1 Obtain the name and partition number of the device that backs the VMFS datastore that you want to check.

   #esxcli storage vmfs extent list

The Device Name and Partition columns in the output identify the device. For example:

```
Volume Name  .....  Device Name                              Partition
1TB_VMFS6    .....  naa.xxxx                                    3
```

2 Check for VMFS errors.

Provide the absolute path to the device partition that backs the VMFS datastore, and provide a partition number with the device name. For example:

voma -m vmfs -f check -d /vmfs/devices/disks/naa.xxxx:x

The output lists possible errors. For example, the following output indicates that the heartbeat address is invalid.

```
XXXXXXXXXXXXXXXXXXXXXXX
Phase 2: Checking VMFS heartbeat region
 ON-DISK ERROR: Invalid HB address
Phase 3: Checking all file descriptors.
Phase 4: Checking pathname and connectivity.
Phase 5: Checking resource reference counts.

Total Errors Found:           1
```

Configuring Pointer Block Cache for vSphere VMFS Datastores

Pointer blocks, also called indirection blocks, are file system resources that contain addresses to VMFS file blocks. When you open a `vmdk` file on an ESXi host, pointer blocks related to that file are stored in the pointer block cache. The size of the pointer block cache is a configurable parameter.

The pointer block cache is a host-wide cache that is independent from VMFS. The cache is shared across all datastores that are accessed from the same ESXi host.

The size of the pointer block cache is controlled by `/VMFS3/MinAddressableSpaceTB` and `/VMFS3/MaxAddressableSpaceTB`. You can configure the minimum and maximum sizes on each ESXi host.

Parameter	Description
/VMFS3/MinAddressableSpaceTB	The minimum value is minimum amount of memory that the system guarantees to the pointer block cache. For example, 1 TB of open file space requires approximately 4 MB of memory. Default value is 10 TB.
/VMFS3/MaxAddressableSpaceTB	The parameter defines the maximum limit of pointer blocks that can be cached in memory. Default value is 32 TB. Maximum value is 128 TB. Typically, the default value of the /VMFS3/MaxAddressableSpaceTB parameter is adequate. However, as the size of the open vmdk files increases, the number of pointer blocks related to those files also increases. If the increase causes any performance degradation, you can adjust the parameter to its maximum value to provide more space for the pointer block cache. Base the maximum size of the pointer block cache on the working set, or the active pointer blocks required.
Pointer Block Eviction	The /VMFS3/MaxAddressableSpaceTB parameter also controls the growth of the pointer block cache. When the size of the pointer block cache approaches the configured maximum size, a pointer block eviction process starts. The mechanism leaves active pointer blocks, but removes non-active or less active blocks from the cache, so that space can be reused.

To change the values for the pointer block cache, use the **Advanced System Settings** dialog box of the vSphere Client or the `esxcli system settings advanced set -o` command.

You can use the `esxcli storage vmfs pbcache` command to obtain information about the size of the pointer block cache and other statistics. This information assists you in adjusting minimum and maximum sizes of the pointer block cache, so that you can get maximum performance.

Obtain Information for VMFS Pointer Block Cache

You can get information about VMFS pointer block cache use. This information helps you understand how much space the pointer block cache consumes. You can also determine whether you must adjust the minimum and maximum sizes of the pointer block cache.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To obtain or reset the pointer block cache statistics, use the following command:

  `esxcli storage vmfs pbcache`

Option	Description
get	Get VMFS pointer block cache statistics.
reset	Reset the VMFS pointer block cache statistics.

Example: Getting Statistics for Pointer Block Cache

```
#esxcli storage vmfs pbcache get
   Cache Capacity Miss Ratio: 0 %
   Cache Size: 0 MiB
   Cache Size Max: 132 MiB
   Cache Usage: 0 %
   Cache Working Set: 0 TiB
   Cache Working Set Max: 32 TiB
   Vmfs Heap Overhead: 0 KiB
   Vmfs Heap Size: 23 MiB
   Vmfs Heap Size Max: 256 MiB
```

Change the Size of the Pointer Block Cache

You can adjust the minimum and maximum sizes of the pointer block cache.

Caution Changing advanced options is considered unsupported. Typically, the default settings produce the optimum result. Change the advanced options only when you get specific instructions from VMware technical support or a knowledge base article.

Procedure

1 Browse to the host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 In Advanced System Settings, select the appropriate item.

Option	Description
VMFS3.MinAddressableSpaceTB	Minimum size of all open files that VMFS cache guarantees to support.
VMFS3.MaxAddressableSpaceTB	Maximum size of all open files that VMFS cache supports before eviction starts.

5 Click the **Edit** button and change the value.

6 Click **OK**.

Example: Use the esxcli Command to Change the Pointer Block Cache

You can also use the `esxcli system settings advanced set -o` to modify the size of the pointer block cache. The following example describes how to set the size to its maximum value of 128 TB.

1 To change the value of `/VMFS3/MaxAddressableSpaceTB` to 128 TB, enter the following command:

```
# esxcli system settings advanced set -i 128 -o /VMFS3/
MaxAddressableSpaceTB
```

2 To confirm that the value is set correctly, enter this command:

```
# esxcli system settings advanced list -o /VMFS3/MaxAddressableSpaceTB
```

NFS Datastore Concepts and Operations in vSphere Environment

An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it for its storage needs. vSphere supports versions 3 and 4.1 of the NFS protocol.

Typically, the NFS volume or directory is created by a storage administrator and is exported from the NFS server. You do not need to format the NFS volume with a local file system, such as VMFS. Instead, you mount the volume directly on the ESXi hosts and use it to store and boot virtual machines in the same way that you use the VMFS datastores.

In addition to storing virtual disks on NFS datastores, you can use NFS as a central repository for ISO images, virtual machine templates, and so on. If you use the datastore for the ISO images, you can connect the CD-ROM device of the virtual machine to an ISO file on the datastore. You then can install a guest operating system from the ISO file.

After you set up NFS storage environment, see the following topics to create and manage NFS datastores:

- Create an NFS Datastore in vSphere Environment
- Basic Administrative Operations for vSphere Datastores

NFS Protocols and ESXi

ESXi supports NFS protocols version 3 and 4.1. To support both versions, ESXi uses two different NFS clients.

Comparing Versions of NFS Clients

The following table lists capabilities that the NFS version 3 and 4.1 support.

Characteristics	NFS version 3	NFS version 4.1
Security mechanisms	AUTH_SYS	AUTH_SYS and Kerberos (krb5 and krb5i)
Encryption algorithms with Kerberos	N/A	AES256-CTS-HMAC-SHA1-96 and AES128-CTS-HMAC-SHA1-96
Multipathing	Not supported	Supported through the session trunking
Locking mechanisms	Propriety client-side locking	Server-side locking
Hardware acceleration	Supported	Supported
Thick virtual disks	Supported	Supported
IPv6	Supported	Supported for AUTH_SYS and Kerberos
ISO images presented as CD-ROMs to virtual machines	Supported	Supported
Virtual machine snapshots	Supported	Supported
Virtual machines with virtual disks greater than 2 TB	Supported	Supported

NFS Protocols and vSphere Solutions

The following table lists major vSphere solutions that NFS versions support.

vSphere Features	NFS version 3	NFS version 4.1
vMotion and Storage vMotion	Yes	Yes
High Availability (HA)	Yes	Yes
Fault Tolerance (FT)	Yes	Yes
Distributed Resource Scheduler (DRS)	Yes	Yes
Host Profiles	Yes	Yes
Storage DRS	Yes	Initial placement and load balancing based on space utilization.
Storage I/O Control	Yes	No
Site Recovery Manager	Yes	Site Recovery Manager does not support NFS 4.1 datastores for array-based replication and Virtual Volumes replication. You can use Site Recovery Manager with NFS v 4.1 datastores for vSphere Replication.
Virtual Volumes	Yes	Yes

vSphere Features	NFS version 3	NFS version 4.1
vSphere Replication	Yes	Yes
VMware Aria Operations	Yes	Yes

NFS 4.1 and Fault Tolerance

Virtual machines on NFS 4.1 support the new Fault Tolerance mechanism that was introduced in vSphere 6.0. The mechanism can accommodate symmetric multiprocessor (SMP) virtual machines with up to four vCPUs.

NFS 4.1 VMs do not support the legacy Fault Tolerance mechanism.

NFS Upgrades

When you upgrade ESXi from a version earlier than 6.5, existing NFS 4.1 datastores automatically begin supporting functionalities that were not available in the previous ESXi release. These functionalities include Virtual Volumes, hardware acceleration, and so on.

ESXi does not support automatic datastore conversions from NFS version 3 to NFS 4.1.

If you want to upgrade your NFS 3 datastore, the following options are available:

- Create the NFS 4.1 datastore, and then use Storage vMotion to migrate virtual machines from the old datastore to the new one.
- Use conversion methods provided by your NFS storage server. For more information, contact your storage vendor.
- Unmount the NFS 3 datastore, and then mount as NFS 4.1 datastore.

  Caution If you use this option, make sure to unmount the datastore from all hosts that have access to the datastore. The datastore can never be mounted by using both protocols at the same time.

Guidelines and Requirements for NFS Storage with ESXi

When you use NFS storage with ESXi, follow specific guidelines related to NFS server configuration, networking, NFS datastores, and so on.

NFS Server Configuration

When you configure NFS servers to work with ESXi, follow recommendation of your storage vendor. In addition to these general recommendations, use specific guidelines that apply to NFS in vSphere environment.

The guidelines include the following items.

- Make sure that the NAS servers you use are listed in the *VMware HCL*. Use the correct version for the server firmware.

- Ensure that the NFS volume is exported using NFS over TCP.
- Make sure that the NAS server exports a particular share as either NFS 3 or NFS 4.1. The NAS server must not provide both protocol versions for the same share. The NAS server must enforce this policy because ESXi does not prevent mounting the same share through different NFS versions.
- NFS 3 and non-Kerberos (AUTH_SYS) NFS 4.1 do not support the delegate user functionality that enables access to NFS volumes using nonroot credentials. If you use NFS 3 or non-Kerberos NFS 4.1, ensure that each host has root access to the volume. Different storage vendors have different methods of enabling this functionality, but typically the NAS servers use the `no_root_squash` option. If the NAS server does not grant root access, you can still mount the NFS datastore on the host. However, you cannot create any virtual machines on the datastore.
- If the underlying NFS volume is read-only, make sure that the volume is exported as a read-only share by the NFS server. Or mount the volume as a read-only datastore on the ESXi host. Otherwise, the host considers the datastore to be read-write and might not open the files.

NFS Networking

An ESXi host uses TCP/IP network connection to access a remote NAS server. Certain guidelines and best practices exist for configuring the networking when you use NFS storage.

For more information, see the *vSphere Networking* documentation.

- For network connectivity, use a standard network adapter in your ESXi host.
- ESXi supports Layer 2 and Layer 3 Network switches. If you use Layer 3 switches, ESXi hosts and NFS storage arrays must be on different subnets and the network switch must handle the routing information.
- Configure a VMkernel port group for NFS storage. You can create the VMkernel port group for IP storage on an existing virtual switch (vSwitch) or on a new vSwitch. The vSwitch can be a vSphere Standard Switch (VSS) or a vSphere Distributed Switch (VDS).
- If you use multiple ports for NFS traffic, make sure that you correctly configure your virtual switches and physical switches.
- NFS 3 and NFS 4.1 support IPv6.
- You can configure NFS storage with multiple connections by using the `nconnect` option. For NFS 4.1, you can create multiple connections per session. For NFS 3, you can mount the datastore with multiple connections. You can set a maximum of 4 connections per NFS datastore by default. However, you can increase it up to 8 by using the advanced NFS option. Ensure that the total number of connections across all mounted NFS datastores does not exceed 256. See Configure Multiple TCP Connections for NFS.

- You can isolate NFS traffic to specific VMkernel adapters. Without binding, if the VMkernel adapter that ESXi uses for NFS traffic fails, the network infrastructure redirects the traffic to an alternative route. As a result, the NFS traffic might unintentionally flow through a random VMkernel adapter. VMkernel port binding for the NFS datastore allows you to bind an NFS volume to a specific VMkernel adapter to connect to an NFS server. See Configure VMkernel Binding for NFS Datastore.

NFS File Locking

File locking mechanisms are used to restrict access to data stored on a server to only one user or process at a time. The locking mechanisms of the two NFS versions are not compatible. NFS 3 uses proprietary locking and NFS 4.1 uses native protocol specified locking.

NFS 3 locking on ESXi does not use the Network Lock Manager (NLM) protocol. Instead, VMware provides its own locking protocol. NFS 3 locks are implemented by creating lock files on the NFS server. Lock files are named `.lck-`*`file_id`*`.`.

NFS 4.1 uses share reservations as a locking mechanism.

Because NFS 3 and NFS 4.1 clients do not use the same locking protocol, you cannot use different NFS versions to mount the same datastore on multiple hosts. Accessing the same virtual disks from two incompatible clients might result in incorrect behavior and cause data corruption.

NFS Security

With NFS 3 and NFS 4.1, ESXi supports the AUTH_SYS security. In addition, for NFS 4.1, the Kerberos security mechanism is supported.

NFS 3 supports the AUTH_SYS security mechanism. With this mechanism, storage traffic is transmitted in an unencrypted format across the LAN. Because of this limited security, use NFS storage on trusted networks only and isolate the traffic on separate physical switches. You can also use a private VLAN.

NFS 4.1 supports the Kerberos authentication protocol to secure communications with the NFS server. Nonroot users can access files when Kerberos is used. For more information, see Using Kerberos for NFS 4.1 with ESXi.

In addition to Kerberos, NFS 4.1 supports traditional non-Kerberos mounts with the AUTH_SYS security. In this case, use root access guidelines for NFS version 3.

Note You cannot use two security mechanisms, AUTH_SYS and Kerberos, for the same NFS 4.1 datastore shared by multiple hosts.

NFS Multipathing

NFS 4.1 supports multipathing as per protocol specifications. For NFS 3 multipathing is not applicable.

NFS 3 uses one TCP connection for I/O. As a result, ESXi supports I/O on only one IP address or hostname for the NFS server, and does not support multiple paths. Depending on your network infrastructure and configuration, you can use the network stack to configure multiple connections to the storage targets. In this case, you must have multiple datastores, each datastore using separate network connections between the host and the storage.

NFS 4.1 provides multipathing for servers that support the session trunking. When the trunking is available, you can use multiple IP addresses to access a single NFS volume. Client ID trunking is not supported.

NFS and Hardware Acceleration

Virtual disks created on NFS datastores are thin-provisioned by default. To be able to create thick-provisioned virtual disks, you must use hardware acceleration that supports the Reserve Space operation.

NFS 3 and NFS 4.1 support hardware acceleration that allows your host to integrate with NAS devices and use several hardware operations that NAS storage provides. For more information, see vSphere Hardware Acceleration on NAS Devices.

NFS Datastores

When you create an NFS datastore, make sure to follow specific guidelines.

The NFS datastore guidelines and best practices include the following items. To create an NFS datastores, see Create an NFS Datastore in vSphere Environment.

- You cannot use different NFS versions to mount the same datastore on different hosts. NFS 3 and NFS 4.1 clients are not compatible and do not use the same locking protocol. As a result, accessing the same virtual disks from two incompatible clients might result in incorrect behavior and cause data corruption.
- NFS 3 and NFS 4.1 datastores can coexist on the same host.
- ESXi cannot automatically upgrade NFS version 3 to version 4.1, but you can use other conversion methods. For information, see NFS Upgrades.
- When you mount the same NFS 3 volume on different hosts, make sure that the server and folder names are identical across the hosts. If the names do not match, the hosts see the same NFS version 3 volume as two different datastores. This error might result in a failure of such features as vMotion. An example of such discrepancy is entering `filer` as the server name on one host and `filer.domain.com` on the other. This guideline does not apply to NFS version 4.1.
- If you use non-ASCII characters to name datastores and virtual machines, make sure that the underlying NFS server offers internationalization support. If the server does not support international characters, use only ASCII characters, or unpredictable failures might occur.

Using Layer 3 Routed Connections to Access NFS Storage with ESXi

When you use Layer 3 (L3) routed connections to access NFS storage, consider certain requirements and restrictions.

Ensure that your environment meets the following requirements:

- Use Cisco's Hot Standby Router Protocol (HSRP) in IP Router. If you are using a non-Cisco router, use Virtual Router Redundancy Protocol (VRRP) instead.
- To prioritize NFS L3 traffic on networks with limited bandwidths, or on networks that experience congestion, use Quality of Service (QoS). See your router documentation for details.
- Follow Routed NFS L3 recommendations offered by storage vendor. Contact your storage vendor for details.
- Deactivate Network I/O Resource Management (NetIORM).
- If you are planning to use systems with top-of-rack switches or switch-dependent I/O device partitioning, contact your system vendor for compatibility and support.

In an L3 environment, the following restrictions apply:

- The environment does not support VMware Site Recovery Manager.
- The environment supports only the NFS protocol. Do not use other storage protocols such as FCoE over the same physical network.
- The NFS traffic in this environment does not support IPv6.
- The NFS traffic in this environment can be routed only over a LAN. Other environments such as WAN are not supported.

Firewall Configurations for NFS Storage ESXi

Learn about firewall rule set files, `nfsClient` and `nfs41client`, that ESXi creates when you mount NFS datatores version 3 or 4.1

For general information about firewall configurations, see Configuring the ESXi Firewall in the *vSphere Security* documentation.

NFS Client Firewall Behavior

The NFS Client firewall rule set behaves differently than other ESXi firewall rule sets. ESXi configures NFS Client settings when you mount or unmount an NFS datastore. The behavior differs for different versions of NFS.

When you add, mount, or unmount an NFS datastore, the resulting behavior depends on the version of NFS.

NFS v3 Firewall Behavior

When you add or mount an NFS v3 datastore, ESXi checks the state of the NFS Client (`nfsClient`) firewall rule set.

- If the `nfsClient` rule set is deactivated, ESXi activates the rule set and deactivates the Allow All IP Addresses policy by setting the `allowedAll` flag to `FALSE`. The IP address of the NFS server is added to the allowed list of outgoing IP addresses.
- If the `nfsClient` rule set is activated, the state of the rule set and the allowed IP address policy are not changed. The IP address of the NFS server is added to the allowed list of outgoing IP addresses.

Note If you manually activate the `nfsClient` rule set or manually set the Allow All IP Addresses policy, either before or after you add an NFS v3 datastore to the system, your settings are overridden when the last NFS v3 datastore is unmounted. The `nfsClient` rule set is deactivated when all NFS v3 datastores are unmounted.

When you remove or unmount an NFS v3 datastore, ESXi performs one of the following actions.

- If none of the remaining NFS v3 datastores are mounted from the server of the datastore being unmounted, ESXi removes the server's IP address from the list of outgoing IP addresses.
- If no mounted NFS v3 datastores remain after the unmount operation, ESXi deactivates the `nfsClient` firewall rule set.

NFS v4.1 Firewall Behavior

When you mount the first NFS v4.1 datastore, ESXi activates the `nfs41client` rule set and sets its `allowedAll` flag to TRUE. This action opens port 2049 for all IP addresses. Unmounting an NFS v4.1 datastore does not affect the firewall state. That is, the first NFS v4.1 mount opens port 2049 and that port remains activated unless you close it explicitly.

Verify Firewall Ports for NFS Clients

To enable access to NFS storage, ESXi automatically opens firewall ports for the NFS clients when you mount an NFS datastore. For troubleshooting reasons, you might need to verify that the ports are open.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **System**, click **Firewall**, and click **Edit**.
4. Scroll down to an appropriate version of NFS to make sure that the port is open.

Using Kerberos for NFS 4.1 with ESXi

With NFS version 4.1, ESXi supports the Kerberos authentication mechanism. The ESXi implementation of Kerberos for NFS 4.1 provides two security models, krb5 and krb5i, that offer different levels of security.

The RPCSEC_GSS Kerberos mechanism is an authentication service. It allows an NFS 4.1 client installed on ESXi to prove its identity to an NFS server before mounting an NFS share. The Kerberos security uses cryptography to work across an insecure network connection.

The ESXi implementation of Kerberos for NFS 4.1 provides two security models, krb5 and krb5i, that offer different levels of security.

- Kerberos for authentication only (krb5) supports identity verification.
- Kerberos for authentication and data integrity (krb5i), in addition to identity verification, provides data integrity services. These services help to protect the NFS traffic from tampering by checking data packets for any potential modifications.

Kerberos supports cryptographic algorithms that prevent unauthorized users from gaining access to NFS traffic. The NFS 4.1 client on ESXi attempts to use either the AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 algorithm to access a share on the NAS server. Before using your NFS 4.1 datastores, make sure that AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 are enabled on the NAS server.

The following table compares Kerberos security levels that ESXi supports.

Type of Kerberos Security		ESXi Support
Kerberos for authentication only (krb5)	Integrity checksum for RPC header	Yes with AES
	Integrity checksum for RPC data	No
Kerberos for authentication and data integrity (krb5i)	Integrity checksum for RPC header	Yes with AES
	Integrity checksum for RPC data	Yes with AES

When you use Kerberos authentication, the following considerations apply:

- ESXi uses Kerberos with the Active Directory domain.
- As a vSphere administrator, you specify Active Directory credentials to provide access to NFS 4.1 Kerberos datastores for an NFS user. A single set of credentials is used to access all Kerberos datastores mounted on that host.
- When multiple ESXi hosts share the NFS 4.1 datastore, you must use the same Active Directory credentials for all hosts that access the shared datastore. To automate the assignment process, set the user in host profiles and apply the profile to all ESXi hosts.
- You cannot use two security mechanisms, AUTH_SYS and Kerberos, for the same NFS 4.1 datastore shared by multiple hosts.

Configure ESXi Hosts for Kerberos Authentication

If you use NFS 4.1 with Kerberos, you must perform several tasks to set up your hosts for Kerberos authentication.

When multiple ESXi hosts share the NFS 4.1 datastore, you must use the same Active Directory credentials for all hosts that access the shared datastore. You can automate the assignment process by setting the user in host profiles and applying the profile to all ESXi hosts.

Prerequisites

- Make sure that Microsoft Active Directory (AD) and NFS servers are configured to use Kerberos.
- Enable AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 encryption modes on AD. The NFS 4.1 client does not support the DES-CBC-MD5 encryption mode.
- Make sure that the NFS server exports are configured to grant full access to the Kerberos user.

Configure DNS for NFS 4.1 with Kerberos

When you use NFS 4.1 with Kerberos, you must change the DNS settings on ESXi hosts. The settings must point to the DNS server that is configured to hand out DNS records for the Kerberos Key Distribution Center (KDC). For example, use the Active Directory server address if AD is used as a DNS server.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Networking**, click **TCP/IP configuration**.

4 Select **Default** and click the **Edit** icon.

5 Manually enter the DNS settings.

Option	Description
Domain	*AD Domain Name*
Preferred DNS server	*AD Server IP*
Search domains	*AD Domain Name*

Configure Network Time Protocol for NFS 4.1 with Kerberos

If you use NFS 4.1 with Kerberos, ESXi hosts, the NFS server, and the Active Domain server need to be time synchronized. Typically, in the setup the Active Domain server is used as the Network Time Protocol (NTP) server.

The following task describes how to synchronize the ESXi host with the NTP server.

The best practice is to use the Active Domain server as the NTP server.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **System**, select **Time Configuration**.
4. Click **Edit** and set up the NTP server.
   a. Select **Use Network Time Protocol (Enable NTP client)**.
   b. To synchronize with the NTP server, enter its IP addresses.
   c. Select **Start NTP Service**.
   d. Set the NTP Service Startup Policy.
5. Click **OK**.

   The host synchronizes with the NTP server.

Enable Kerberos Authentication in Active Directory

If you use NFS 4.1 storage with Kerberos, you must add each ESXi host to an Active Directory domain and enable Kerberos authentication. Kerberos integrates with Active Directory to enable single sign-on and provides an extra layer of security when used across an insecure network connection.

Prerequisites

Set up an AD domain and a domain administrator account with the rights to add hosts to the domain.

Procedure

1. In the vSphere Client, navigate to the ESXi host.
2. Click the **Configure** tab.
3. Under **System**, click **Authentication Services**.
4. Add the ESXi host to an Active Directory domain.
   a. In the Authentication Services pane, click **Join Domain**.
   b. Supply the domain settings, and click **OK**.

   The directory services type changes to Active Directory.
5. Configure or edit credentials for an NFS Kerberos user.
   a. In the NFS Kerberos Credentials pane, click **Edit**.
   b. Enter a user name and password.

      Files stored in all Kerberos datastores are accessed using these credentials.

   The state for NFS Kerberos credentials changes to Enabled.

What to do next

After you configure your host for Kerberos, you can create an NFS 4.1 datastore with Kerberos enabled.

For information about creating an NFS user account to be used by the NFS 4.1 for Kerberos authentication, see VMware PowerCLI on the Broadcom Developer Portal page at https://developer.broadcom.com/powercli/latest/vmware.vimautomation.storage/commands/new-nfsuser.

Set Up NFS Storage Environment with ESXi

You must perform several configuration steps before you mount an NFS datastore in vSphere.

Prerequisites

- Familiarize yourself with the guidelines in Guidelines and Requirements for NFS Storage with ESXi.
- For details on configuring NFS storage, consult your storage vendor documentation.
- If you use Kerberos, make sure that AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 are enabled on the NAS server.

Procedure

1. On the NFS server, configure an NFS volume and export it to be mounted on the ESXi hosts.

   a. Note the IP address or the DNS name of the NFS server and the full path, or folder name, for the NFS share.

      For NFS 4.1, you can collect multiple IP addresses or DNS names to use the multipathing support that the NFS 4.1 datastore provides.

   b. If you plan to use Kerberos authentication with NFS 4.1, specify the Kerberos credentials to be used by ESXi for authentication.

2. On each ESXi host, configure a VMkernel Network port for NFS traffic.

   For more information, see the *vSphere Networking* documentation.

3. If you plan to use Kerberos authentication with the NFS 4.1 datastore, configure the ESXi hosts for Kerberos authentication.

   See Configure ESXi Hosts for Kerberos Authentication.

What to do next

You can now create an NFS datastore on the ESXi hosts. See Create an NFS Datastore in vSphere Environment.

Configuring the NFS Datastore

You can use the esxcli command or the vSphere Client to configure the NFS datastore depending on the NFS version and the configuration you want to set.

Understand the esxcli Command Parameters for NFS Datastores

Use the esxcli command to configure specific settings on NFS.

Command	Description
NFS 4.1: esxcli storage nfs41 add NFS 3: esxcli storage nfs add	Adds a new NFS or NAS volume and mounts it with the given volume name on the ESXi host .
NFS 4.1: esxcli storage nfs41 param set NFS 3: esxcli storage nfs param set	Sets the NFS volume parameters.

The commands accept the following options:

Option	Description
-H|--hosts=<str>	NFS 4.1: The hostname or IP address of the server for the NFS 4.1 volume you want to add. You can add multiple NFS volumes separated with a comma. NFS 3: The hostname or IP address of the NAS volume you want to add and mount on the system.
-I|--host-vmknic=<str>	NFS 4.1: The hostname or IP address of the server for the NFS 4.1 volume you want to add followed by a colon(:) and the VMknic name. You can specify multiple names. Example: --host-vmknic=<host1>:<vmknic1> --host-vmknic=<ip2>:<vmknic2> NFS 3: The hostname or IP address of the NAS volume you want to add. You can specify only one name. Example: <ip1>:<vmknic1>
-p|--ispe	If set, this flag will set PE as the mount point.
-r|--readonly	If set, this flag will make the mount point be read-only.
-a|--sec=<str>	Security setting. Acceptable values are: AUTH_SYS, SEC_KRB5, and SEC_KRB5I. This option is applicable only for NFS 4.1 and later.
-s|--share=<str>	The share name to use for the NFS or NAS mount point on the remote system. This option is required.
-v|--volume-name=<str>	The volume name to use for the NFS or NAS mount. The volume name must be unique and should not conflict with existing NAS, NFS, VMFS, or other volume names. This option is required.

Option	Description
`-c|--connections=<long>`	The number of RPC connections to be established for the NFS datastore. A higher number of connections results in better performance on datastores with higher I/O load.
`-q|--maxqueuedepth=<long>`	The maximum queue depth for the NFS volume. This option can be used only with the `param set` command.

Configure Multiple TCP Connections for NFS

You can configure NFS storage with multiple connections by using the `nconnect` option. For NFS 4.1, you can create multiple connections per session. For NFS 3, you can mount the datastore with multiple connections. Use the `esxcli storage nfs41` command to configure multiple connections.

You can set a maximum of 4 connections per NFS datastore by default. However, you can increase it up to 8 by using the advanced NFS option. Ensure that the total number of connections across all mounted NFS datastores does not exceed 256.

For NFS 4.1, setting up multiple connections does not impact multipathing. Multiple connections and multiple paths can co-exist.

Note The commands in the folllowing procedure are examples for NFS 4.1 datastores. For NFS 3 datastores, replace `nfs41` with `nfs`.

Prerequisites

Set up the NFS storage environment.

Procedure

1 To specify the number of connections while mounting the NFS datastore, use the following command:

```
esxcli storage nfs41 add -H <host> -v <volume-label> -s <remote_share> -c
<number_of_connections>
```

2 To specify the number of connections on an existing NFS datastore, use the following command:

```
esxcli storage nfs41 param set -v <volume-label> -c
<number_of_connections>
```

3 (Optional) To increase the number of connections up to 8, use the following command:

```
esxcli storage nfs41 param set -v <volume-label> -c
<number_of_connections>
```

Configure VMkernel Binding for NFS Datastore

VMkernel port binding for the NFS datastore on an ESXi host allows you to bind an NFS volume to a specific VMkernel adapter to connect to an NFS server. With NFS datastores, you can

isolate NFS traffic to a specific VMkernel adapter. To route the traffic to this adapter, connect the datastore to the adapter.

Note Without binding, if the VMkernel adapter that ESXi uses for NFS traffic fails, the network infrastructure redirects the traffic to an alternative route. As a result, the NFS traffic might unintentionally flow through a random VMkernel adapter.

Configure VMkernel Binding for NFS 4.1 Datastores

To connect an NFS 4.1 datastore to a specific VMkernel adapter, use the `esxcli storage nfs41` command. You can specify the VMkernel adapters either while mounting the NFS 4.1 datastore or, if the NFS datastore is already mounted with a specific VMkernel adapter, you can modify the existing NFS volume configuration to specify the desired VMkernel adapters.

Prerequisites

- Set up the NFS storage environment.
- Create and configure a VMkernel adapter on default TCP/IP stack.

Procedure

1 To specify VMkernel adapters while mounting a new NFS v4.1 volume on the ESXi host, use the following command:

   `[root@:~] esxcli storage nfs41 add`

   Example: `[root@:~]esxcli storage nfs41 add -I 10.187.114.32:vmk0 -I 10.187.214.27:vmk1 -s /store1 -v store1`

2 To set the VMkernel adapters on an existing NFS 4.1 datastore, use the following command:

   `[root@:~] esxcli storage nfs41 param set`

   Example: `[root@:~] esxcli storage nfs41 param set -I 10.187.114.32:vmk2 -v store1`

3 (Optional) Verify whether the specified VMkernel adapters are set:

   `[root@:~] esxcli storage nfs41 list`

   The output should be similar to the following:

```
Volume Name  Host(s)                      Share    Vmknics      Accessible  Mounted
Connections  Read-Only  Security   isPE  Hardware Acceleration
-----------  -------------                -------  -------      ----------  -------
-----------  ---------  --------  -----  ---------------------
store1       10.187.114.32,10.187.214.27   /store1   vmk2,vmk1        true
true            1                         false              AUTH_SYS  false     Not
Supported
```

Configure VMkernel Binding for NFS 3 Datastores

For NFS 3 datastores, you can use the vSphere Client to connect the datastore to the desired VMkernel adapters.

Prerequisites

- Set up the NFS storage environment.
- Create and configure a VMkernel adapter on default TCP/IP stack.

Procedure

1 Bind an NFS 3 datastore to a VMkernel adapter.

   a In the vSphere Client navigator, browse to a host, a cluster, or a data center.

   b From the right-click menu, select **Storage > New Datastore**.

   c Select **NFS** as the datastore type, and select **NFS 3**.

   d Enter the datastore parameters. For more information, see Create an NFS Datastore in vSphere Environment.

   e Select **Bind to vmknic**.

   f If you are creating the datastore at the data center or cluster level, select hosts that mount the datastore.

   g For the datastore at the data center or cluster level, select one of the following options.

   - Select **Vmknics on all hosts**. With this option, selectable VMkernel adapters are available across all hosts where you mount your datastore.
   - Select **Vmknic per host**. With this option, you can specify a VMkernel adapter per host.

   Note You can perform only one selection at a time.

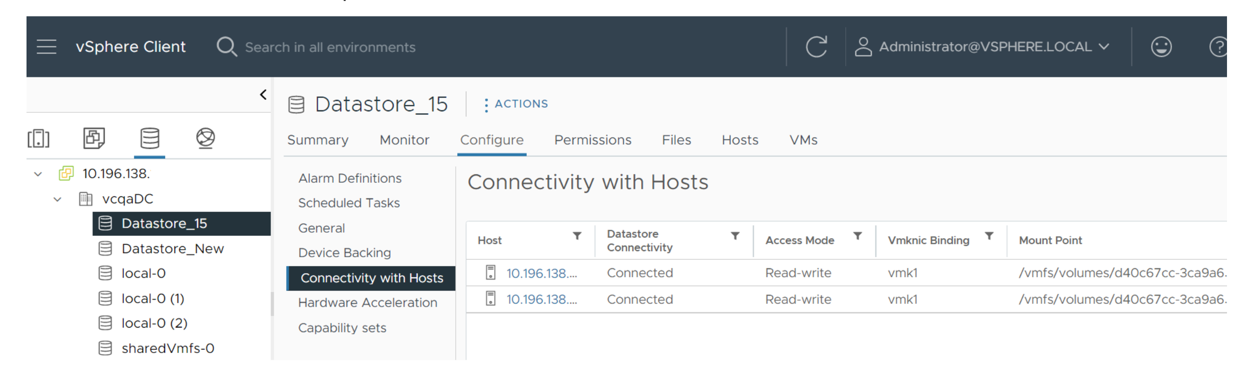
   h Review the configuration options you made and click **Finish**.

2 Review the hosts connected to the NFS 3 datastore.

   a In the vSphere Client, select the NFS 3 datastore with VMkernel adapter binding.

   b Click the **Configure** tab, and click **Connectivity with Hosts**.

You can view all the hosts connected to the NFS 3 datastore. The **Vmknic Binding** column shows the VMkernel adapters connected to the datastore



Note If you delete the VMkernel adapter associated with the NFS 3 datastore, the datastore becomes inaccessible triggering a warning. The warning displays the text in the **Events** screen similar to the following.

```
vmknic vmkX removed, NFSv3 datastore datastore_name configured with the vmknic will
be inaccessible.
```

Collecting Statistical Information for NFS Storage on ESXi Host

You can use the `nfsStats` tool in your ESXi host to display statistical information about NFS calls and Remote Procedure Calls (RPC). The command displays statistical information for NFS 3 and NFS 4.1 mounts on the ESXi host.

Generally, the `nfsStats` tool performs the following tasks.

- Gathers NFS statistics to investigate problems when you deploy a new configuration, such as a new NFS server or network, in the NFS environment.
- Provides statistics about the success and failure of NFS operations.
- Publishes latency statistics about the success and failure of NFS operations.
- Troubleshoots NFS performance issues.

The command syntax is `nfsStats` *options*.

The following command options are available.

Table 9-7. `nfsStats` **commands**

Command Option	Description
`No option`	Obtain both NFS statistics and RPC statistics for all NFS datastores.
`-3`	Display only NFS 3 statistics.
`-4`	Display only NFS 4.1 statistics.
`-n`	Display only NFS 3 and NFS 4.1 statistics.

Table 9-7. `nfsStats` commands (continued)

Command Option	Description
`-r`	Display RPC statistics.
`-i` *interval*	Display NFS and RPC statistics at an interval equal to the specified value in seconds. For example, if the value you enter is 10, the statistics is refreshed every 10 seconds.
`-j`	Display statistics in json format.
`-e`	Display error statistics for NFS 3, NFS 4.1, and RPC protocols.
`-d, --datastore-name` *DSNAME1, DSNAME2, ...*	Display both NFS and RPC statistics for the specified NFS datastores. Use this option in conjunction with the type of the NFS datastore, for example, -3 or -4.

Using Datastore Browser in vSphere Environment

Use the datastore file browser in your vSphere environment to manage contents of your datastores. You can upload files or folders, download files from the datastore, or move or copy folders or files to a new location. You can also use the datastore browser to inflate the thin virtual disk.

Navigate to the Datastore Browser

Navigate to your datastore to browse folders and files that are stored on the datastore and perform administrative tasks on your folders and files.

Procedure

1 Open the datastore browser.

   a Display the datastore in the inventory.

   b Right-click the datastore and select **Browse Files**.

2 Explore the contents of the datastore by navigating to existing folders and files.

3 Perform administrative tasks by using the icons and options.

Icons and Options	Descriptions
Upload Files	Upload a file to the datastore.
Upload Folder (Available only in the vSphere Client)	Upload a folder to the datastore.
Download	Download from the datastore.
New Folder	Create a folder on the datastore.
Copy to	Copy selected folders or files to a new location, either on the same datastore or on a different datastore.

Icons and Options	Descriptions
Move to	Move selected folders or files to a new location, either on the same datastore or on a different datastore.
Rename to	Rename selected files.
Delete	Delete selected folders or files.
Inflate	Convert a selected thin virtual disk to thick. This option applies only to thin-provisioned disks.

Upload Files or Folders to Datastores

Use the datastore file browser to upload files to datastores on the ESXi host. If you use the vSphere Client, you can also upload folders.

In addition to their traditional use as storage for virtual machines files, datastores can serve to store data or files related to virtual machines. For example, you can upload ISO images of operating systems from a local computer to a datastore on the host. You then use these images to install guest operating systems on the new virtual machines.

Note You cannot upload files directly to the Virtual Volumes datastores. You must first create a folder on the Virtual Volumes datastore, and then upload the files into the folder. The created folders in Virtual Volumes datastores for block storage have a limited storage capacity space of 4GB. The Virtual Volumes datastore supports direct uploads of folders.

Prerequisites

Required privilege: **Datastore.Browse Datastore** and **Host.System Management**

Procedure

1 Open the datastore browser.

   a Display the datastore in the inventory.

   b Right-click the datastore and select **Browse Files**.

2 (Optional) Create a folder to store the file or folder.

3 Upload the file or folder.

Option	Description
Upload a file	a Select the target folder and click **Upload Files**. b Locate the item to upload on the local computer and click **Open**.
Upload a folder (available only in the vSphere Client)	a Select the datastore or the target folder and click **Upload Folders**. b Locate the item to upload on the local computer and click **Ok**.

4 Refresh the datastore file browser to see the uploaded files or folders on the list.

What to do next

You might experience problems when deploying an OVF template that you previously exported and then uploaded to a datastore. For details and a workaround, see the VMware Knowledge Base article 2117310.

Download Files from Datastores

Use the datastore file browser to download files from the datastore available on your ESXi host to your local computer.

Prerequisites

- Required privilege: **Datastore.Browse Datastore**
- Privileges when a VM is encrypted: **Cryptographic operations.Decrypt** and **Cryptographic operations.Direct Access**

Procedure

1 Open the datastore browser.

   a Display the datastore in the inventory.

   b Right-click the datastore and select **Browse Files**.

2 Navigate to the file to download and click **Download**.

3 Follow the prompts to save the file to your local computer.

Rename Datastore Files

Use the datastore browser to rename files.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

1 Open the datastore browser.

   a Display the datastore in the inventory.

   b Right-click the datastore and select **Browse Files**.

2 Browse to a file you want to rename.

3 Select the file and click **Rename to**.

4 Specify the new name and click **OK**.

Move or Copy Datastore Folders or Files

Use the datastore browser to move or copy folders or files to a new location, either on the same datastore or on a different datastore.

Note Virtual disk files are moved or copied without format conversion. If you move a virtual disk to a datastore that belongs to a host different from the source host, you might need to convert the virtual disk. Otherwise, you might not be able to use the disk.

You cannot copy VM files across vCenter Servers.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

1 Open the datastore browser.

   a Display the datastore in the inventory.

   b Right-click the datastore and select **Browse Files**.

2 Browse to an object you want to move or copy, either a folder or a file.

3 Select the object and click **Move to** or **Copy to**.

4 Specify the destination location.

5 (Optional) Select **Overwrite files and folders with matching names at the destination**.

6 Click **OK**.

Inflate Thin Virtual Disks

If you created a virtual disk in the thin format, you can change the format to thick.

You use the datastore browser to inflate the thin virtual disk.

Prerequisites

- Make sure that the datastore where the virtual machine resides has enough space.
- Make sure that the virtual disk is thin.
- Remove snapshots.
- Power off your virtual machine.

Procedure

1 In the vSphere Client, navigate to the folder of the virtual disk you want to inflate.

   a Navigate to the virtual machine.

   b Click the **Datastores** tab.

      The datastore that stores the virtual machine files is listed.

   c Right-click the datastore and select **Browse Files**.

      The datastore browser displays contents of the datastore.

2 Expand the virtual machine folder and browse to the virtual disk file that you want to convert.

   The file has the `.vmdk` extension and is marked with the virtual disk ( ) icon.

3 Select the virtual disk file and click **Inflate**.

   Note The option might not be available if the virtual disk is thick or when the virtual machine is running.

Results

The inflated virtual disk occupies the entire datastore space originally provisioned to it.

Understanding Multipathing and Failover in the ESXi Environment

10

To maintain a constant connection between a host and its storage, ESXi supports multipathing. With multipathing, you can use more than one physical path that transfers data between the host and an external storage device.

If a failure of any element in the SAN network, such as an adapter, switch, or cable, occurs, ESXi can switch to another viable physical path. This process of path switching to avoid failed components is known as path failover.

In addition to path failover, multipathing provides load balancing. Load balancing is the process of distributing I/O loads across multiple physical paths. Load balancing reduces or removes potential bottlenecks.

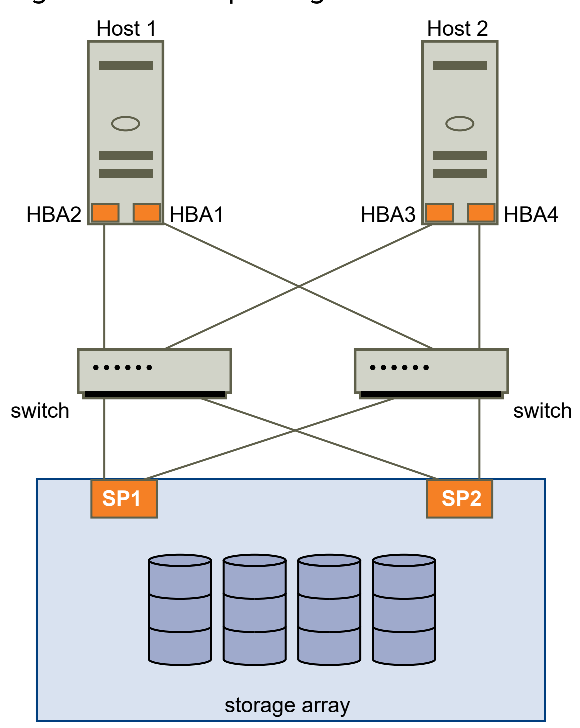
Note Virtual machine I/O might be delayed for up to 60 seconds while path failover takes place. With these delays, the SAN can stabilize its configuration after topology changes. In general, the I/O delays might be longer on active-passive arrays and shorter on active-active arrays.

Failover with Fibre Channel

To support multipathing, your host typically has two or more HBAs available. This configuration supplements the SAN multipathing configuration. Generally, the SAN multipathing provides one or more switches in the SAN fabric and one or more storage processors on the storage array device itself.

In the following illustration, multiple physical paths connect each server with the storage device. For example, if HBA1 or the link between HBA1 and the FC switch fails, HBA2 takes over and provides the connection. The process of one HBA taking over for another is called HBA failover.

Figure 10-1. Multipathing and Failover with Fibre Channel



Similarly, if SP1 fails or the links between SP1 and the switches breaks, SP2 takes over. SP2 provides the connection between the switch and the storage device. This process is called SP failover. VMware ESXi supports both HBA and SP failovers.

Host-Based Failover with iSCSI

When setting up your ESXi host for multipathing and failover, you can use multiple iSCSI HBAs or combine multiple NICs with the software iSCSI adapter.

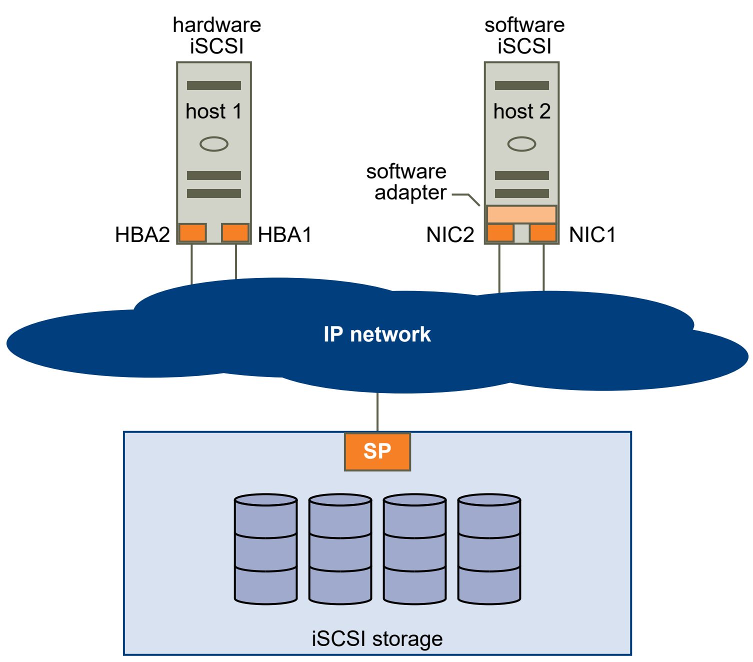
For information on different types of iSCSI adapters, see iSCSI Initiators.

When you use multipathing, specific considerations apply.

- ESXi does not support multipathing when you combine an independent hardware adapter with software iSCSI or dependent iSCSI adapters in the same host.
- Multipathing between software and dependent adapters within the same host is supported.
- On different hosts, you can mix both dependent and independent adapters.

The following illustration shows multipathing setups possible with different types of iSCSI initiators.

Figure 10-2. Host-Based Path Failover



Hardware iSCSI and Failover

With hardware iSCSI, the host typically has two or more hardware iSCSI adapters. The host uses the adapters to reach the storage system through one or more switches. Alternatively, the setup might include one adapter and two storage processors, so that the adapter can use different paths to reach the storage system.

On the illustration, Host1 has two hardware iSCSI adapters, HBA1 and HBA2, that provide two physical paths to the storage system. Multipathing plug-ins on your host, whether the VMkernel NMP or any third-party MPPs, have access to the paths by default. The plug-ins can monitor health of each physical path. If, for example, HBA1 or the link between HBA1 and the network fails, the multipathing plug-ins can switch the path over to HBA2.

Software iSCSI and Failover

With software iSCSI, as shown on Host 2 of the illustration, you can use multiple NICs that provide failover and load balancing capabilities for iSCSI connections.

Multipathing plug-ins do not have direct access to physical NICs on your host. As a result, for this setup, you first must connect each physical NIC to a separate VMkernel port. You then associate all VMkernel ports with the software iSCSI initiator using a port binding technique. Each VMkernel port connected to a separate NIC becomes a different path that the iSCSI storage stack and its storage-aware multipathing plug-ins can use.

For information about configuring multipathing for software iSCSI, see Setting Up Network for iSCSI and iSER with ESXi.

Array-Based Failover with iSCSI

Some iSCSI storage systems manage path use of their ports automatically and transparently to ESXi.

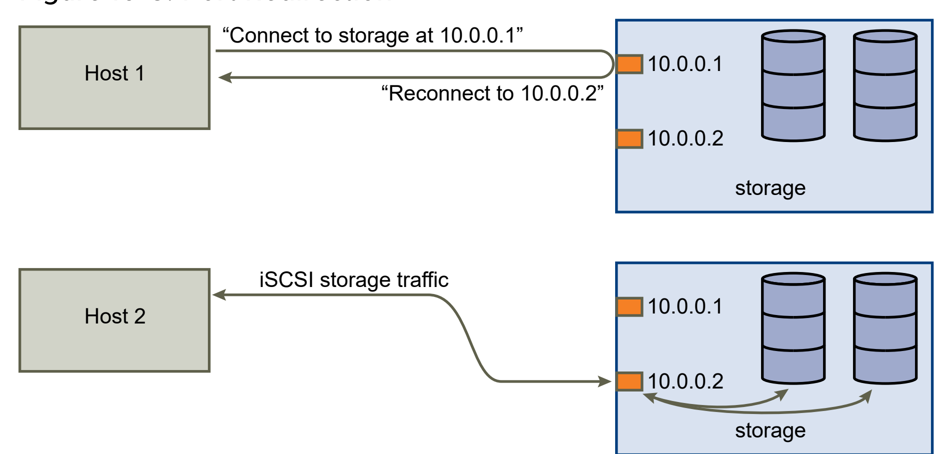
When using one of these storage systems, your host does not see multiple ports on the storage and cannot choose the storage port it connects to. These systems have a single virtual port address that your host uses to initially communicate. During this initial communication, the storage system can redirect the host to communicate with another port on the storage system. The iSCSI initiators in the host obey this reconnection request and connect with a different port on the system. The storage system uses this technique to spread the load across available ports.

If the ESXi host loses connection to one of these ports, it automatically attempts to reconnect with the virtual port of the storage system, and should be redirected to an active, usable port. This reconnection and redirection happens quickly and generally does not disrupt running virtual machines. These storage systems can also request that iSCSI initiators reconnect to the system, to change which storage port they are connected to. This allows the most effective use of the multiple ports.

The Port Redirection illustration shows an example of port redirection. The host attempts to connect to the 10.0.0.1 virtual port. The storage system redirects this request to 10.0.0.2. The host connects with 10.0.0.2 and uses this port for I/O communication.

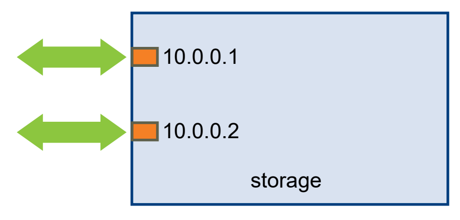
Note The storage system does not always redirect connections. The port at 10.0.0.1 could be used for traffic, also.

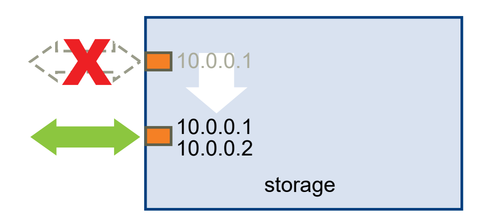
Figure 10-3. Port Redirection



If the port on the storage system that is acting as the virtual port becomes unavailable, the storage system reassigns the address of the virtual port to another port on the system. Port Reassignment shows an example of this type of port reassignment. In this case, the virtual port 10.0.0.1 becomes unavailable and the storage system reassigns the virtual port IP address to a different port. The second port responds to both addresses.

Figure 10-4. Port Reassignment





With this form of array-based failover, you can have multiple paths to the storage only if you use multiple ports on the ESXi host. These paths are active-active. For additional information, see Managing iSCSI Session on ESXi Host.

Path Failover and Virtual Machines

When a path fails, storage I/O might pause for 30-60 seconds until your host determines that the link is unavailable and performs the failover. If you attempt to display the host, its storage devices, or its adapters, the operation might appear to stall. Virtual machines with their disks installed on the SAN can appear unresponsive. After the failover, I/O resumes normally and the virtual machines continue to run.

A Windows virtual machine might interrupt the I/O and eventually fail when failovers take too long. To avoid the failure, set the disk timeout value for the Windows virtual machine to at least 60 seconds.

To avoid disruptions during a path failover, increase the standard disk timeout value on a Windows guest operating system.

This procedure explains how to change the timeout value by using the Windows registry.

Tip Before you begin: Back up the Windows registry.

---

1 Select **Start** > **Run**.

2 Type **`regedit.exe`**, and click **OK**.

3 In the left-panel hierarchy view, double-click **HKEY_LOCAL_MACHINE** > **System** > **CurrentControlSet** > **Services** > **Disk**.

4 Double-click **TimeOutValue**.

5 Set the value data to 0x3c (hexadecimal) or 60 (decimal) and click **OK**.

   After you make this change, Windows waits at least 60 seconds for delayed disk operations to finish before it generates errors.

6 Reboot guest OS for the change to take effect.

Read the following topics next:

- Using Pluggable Storage Architecture and Path Management with ESXi
- VMware Native Multipathing Plug-In
- VMware Path Selection Plug-Ins and Policies
- VMware SATPs
- VMware High Performance Plug-In and Path Selection Schemes
- Viewing and Managing Storage Paths on ESXi Hosts
- Using Claim Rules to Control ESXi Multipathing Modules
- Scheduling Queues for Virtual Machine I/Os in vSphere Environment

Using Pluggable Storage Architecture and Path Management with ESXi

To manage multipathing, ESXi uses a special VMkernel layer, Pluggable Storage Architecture (PSA). The PSA is an open and modular framework that coordinates various software modules responsible for multipathing operations, VMware NMP, VMware HPP, and third-party MPPs.

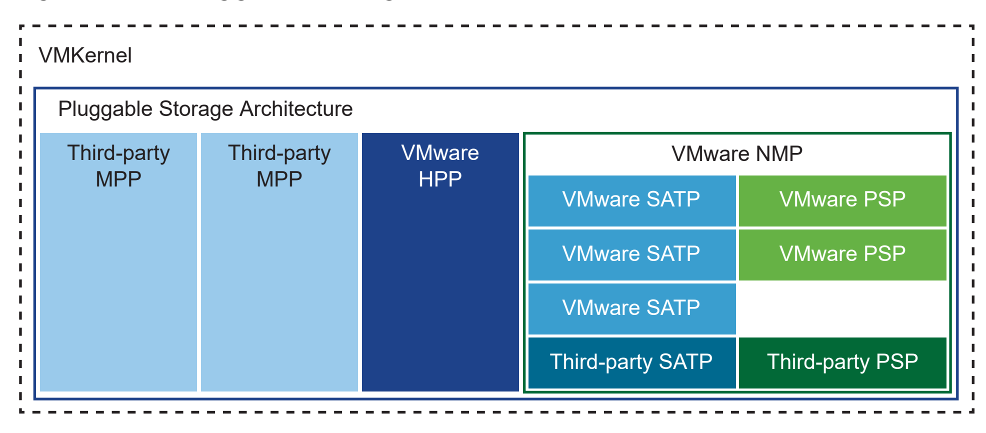
VMware provides generic native multipathing modules, called VMware NMP and VMware HPP. In addition, the PSA offers a collection of VMkernel APIs that third-party developers can use. The software developers can create their own load balancing and failover modules for a particular storage array. These third-party multipathing modules (MPPs) can be installed on the ESXi host and run in addition to the VMware native modules, or as their replacement.

When coordinating the VMware native modules and any installed third-party MPPs, the PSA performs the following tasks:

- Loads and unloads multipathing plug-ins.
- Hides virtual machine specifics from a particular plug-in.
- Routes I/O requests for a specific logical device to the MPP managing that device.
- Handles I/O queueing to the logical devices.
- Implements logical device bandwidth sharing between virtual machines.
- Handles I/O queueing to the physical storage HBAs.
- Handles physical path discovery and removal.
- Provides logical device and physical path I/O statistics.

As the Pluggable Storage Architecture illustration shows, multiple third-party MPPs can run in parallel with the VMware NMP or HPP. When installed, the third-party MPPs can replace the behavior of the native modules. The MPPs can take control of the path failover and the load-balancing operations for the specified storage devices.

Figure 10-5. Pluggable Storage Architecture



Multipathing Concepts and Acronyms

Multipathing Concept	Acronym	Definition
Pluggable Storage Architecture	PSA	The PSA is an open and modular framework that coordinates various software modules responsible for multipathing operations. These modules include generic multipathing modules that VMware provides, NMP and HPP, and third-party MPPs.
Native Multipathing Plug-in	NMP	The NMP is the VMkernel multipathing module that ESXi provides by default and is used for SCSI storage devices. The NMP associates physical paths with a specific storage device and provides a default path selection algorithm based on the array type. The NMP is extensible and manages additional submodules, called Path Selection Policies (PSPs) and Storage Array Type Policies (SATPs). PSPs and SATPs can be provided by VMware, or by a third party. For more information, see VMware Native Multipathing Plug-In.
Path Selection Plug-in	PSP	The PSPs are submodules of the VMware NMP. PSPs are responsible for selecting a physical path for SCSI I/O requests. For more information, see VMware Path Selection Plug-Ins and Policies.
Storage Array Type Plug-in	SATP	SATPs are responsible for array-specific operations for a SCSI storage array. The SATP can determine the state of a particular array-specific path, perform a path activation, and detect any path errors. For more information, see VMware SATPs.
Multipathing Plug-in	MPP (third-party)	The PSA offers a collection of VMkernel APIs that third parties can use to create their own multipathing plug-ins (MPPs). The modules provide specific load balancing and failover functionalities for a particular storage array. The MPPs can be installed on the ESXi host. They can run in addition to the VMware native modules, or as their replacement.

Multipathing Concept	Acronym	Definition
Native High-Performance Plug-in	HPP	The HPP replaces the NMP for high-speed devices, such as NVMe. The HPP can improve the performance of ultra-fast flash devices that are installed locally on your ESXi host, and is the default plug-in that claims NVMe-oF targets. For information, see VMware High Performance Plug-In and Path Selection Schemes.
Path Selection Scheme	PSS	To support multipathing, the HPP uses the Path Selection Schemes (PSS). A particular PSS is responsible for selecting physical paths for I/O requests and handles multipathing for NVMe storage devices. For information, see VMware High Performance Plug-In and Path Selection Schemes.
Claim Rules		The PSA uses claim rules to determine which plug-in owns the paths to a particular storage device. See Using Claim Rules to Control ESXi Multipathing Modules.

VMware Native Multipathing Plug-In

By default, ESXi provides an extensible multipathing module called Native Multipathing Plug-In (NMP).

Generally, the VMware NMP supports all storage arrays listed on the VMware storage HCL and provides a default path selection algorithm based on the array type. The NMP associates a set of physical paths with a specific storage device, or LUN.

For additional multipathing operations, the NMP uses submodules, called SATPs and PSPs. The NMP delegates to the SATP the specific details of handling path failover for the device. The PSP handles path selection for the device.

Typically, the NMP performs the following operations:

- Manages physical path claiming and unclaiming.
- Registers and de-registers logical devices.
- Associates physical paths with logical devices.
- Supports path failure detection and remediation.
- Processes I/O requests to logical devices:
  - Selects an optimal physical path for the request.
  - Performs actions necessary to handle path failures and I/O command retries.
- Supports management tasks, such as reset of logical devices.

ESXi automatically installs an appropriate SATP for an array you use. You do not need to obtain or download any SATPs.

VMware NMP Flow of I/O

When a virtual machine issues an I/O request to a storage device managed by the NMP, the following process takes place.

1 The NMP calls the PSP assigned to this storage device.

2 The PSP selects an appropriate physical path on which to issue the I/O.

3 The NMP issues the I/O request on the path selected by the PSP.

4 If the I/O operation is successful, the NMP reports its completion.

5 If the I/O operation reports an error, the NMP calls the appropriate SATP.

6 The SATP interprets the I/O command errors and, when appropriate, activates the inactive paths.

7 The PSP is called to select a new path on which to issue the I/O.

Display Multipathing Modules

Use the `esxcli` command to list all multipathing modules loaded into the system. Multipathing modules manage physical paths that connect your host with storage. The modules include VMware native NMP and HPP, and any third-party MPPs.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

◆ To list multipathing modules, run the following command:

  `esxcli storage core plugin list --plugin-class=MP`

Results

This command typically shows the NMP and, if loaded, the HPP and the MASK_PATH module. If any third-party MPPs have been loaded, they are listed as well.

```
Plugin name  Plugin class
-----------  ------------
NMP          MP
```

For more information about the command, see the *ESXCLI Concepts and Examples* and *ESXCLI Reference* documentation.

Display NMP Storage Devices

Use the `esxcli` command to list all storage devices controlled by the VMware NMP and display SATP and PSP information associated with each device.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To list all storage devices, run the following command:

  `esxcli storage nmp device list`

  Use the **`--device | -d=`*`device_ID`*** parameter to filter the output of this command to show a single device.

Example: Displaying NMP Storage Devices

```
# esxcli storage nmp device list
mpx.vmhba1:C0:T2:L0
   Device Display Name: Local VMware Disk (mpx.vmhba1:C0:T2:L0)
   Storage Array Type: VMW_SATP_LOCAL
   Storage Array Type Device Config: SATP VMW_SATP_LOCAL does not support device
configuration.
   Path Selection Policy: VMW_PSP_FIXED
   Path Selection Policy Device Config: {preferred=vmhba1:C0:T2:L0;current=vmhba1:C0:T2:L0}
   Path Selection Policy Device Custom Config:
   Working Paths: vmhba1:C0:T2:L0
   Is USB: false

......

eui.6238666462643332
   Device Display Name: SCST_BIO iSCSI Disk (eui.6238666462643332)
   Storage Array Type: VMW_SATP_DEFAULT_AA
   Storage Array Type Device Config: {action_OnRetryErrors=off}
   Path Selection Policy: VMW_PSP_FIXED
   Path Selection Policy Device Config: {preferred=vmhba65:C0:T0:L0;current=vmhba65:C0:T0:L0}
   Path Selection Policy Device Custom Config:
   Working Paths: vmhba65:C0:T0:L0
   Is USB: false
```

For more information about the command, see the *ESXCLI Concepts and Examples* and *ESXCLI Reference* documentation.

VMware Path Selection Plug-Ins and Policies

VMware Path Selection Plug-ins (PSPs) are responsible for selecting a physical path for I/O requests.

The plug-ins are submodules of the VMware NMP. The NMP assigns a default PSP for each logical device based on the device type. You can override the default PSP. For more information, see Change the Path Selection Policy.

Each PSP enables and enforces a corresponding path selection policy.

VMW_PSP_MRU - Most Recently Used (VMware)

The Most Recently Used (VMware) policy is enforced by VMW_PSP_MRU. It selects the first working path discovered at system boot time. When the path becomes unavailable, the host selects an alternative path. The host does not revert to the original path when that path becomes available. The Most Recently Used policy does not use the preferred path setting. This policy is default for most active-passive storage devices.

The VMW_PSP_MRU supports path ranking. To set ranks to individual paths, use the `esxcli storage nmp psp generic pathconfig set` command. For details, see the VMware knowledge base article at http://kb.vmware.com/kb/2003468 and the *ESXCLI Reference* documentation.

VMW_PSP_FIXED - Fixed (VMware)

This Fixed (VMware) policy is implemented by VMW_PSP_FIXED. The policy uses the designated preferred path. If the preferred path is not assigned, the policy selects the first working path discovered at system boot time. If the preferred path becomes unavailable, the host selects an alternative available path. The host returns to the previously defined preferred path when it becomes available again.

Fixed is the default policy for most active-active storage devices.

VMW_PSP_RR - Round Robin (VMware)

VMW_PSP_RR enables the Round Robin (VMware) policy. Round Robin is the default policy for many arrays. It uses an automatic path selection algorithm rotating through the configured paths.

Both active-active and active-passive arrays use the policy to implement load balancing across paths for different LUNs. With active-passive arrays, the policy uses active paths. With active-active arrays, the policy uses available paths.

The latency mechanism that can be activated for the policy makes it more adaptive. To achieve better load balancing results, the mechanism dynamically selects an optimal path by considering the following path characteristics:

- I/O bandwidth
- Path latency

To change the default parameters for the adaptive latency Round Robin policy or to deactivate the latency mechanism, see the Change Default Parameters for Latency Round Robin.

To set other configurable parameters for VMW_PSP_RR, use the `esxcli storage nmp psp roundrobin` command. For details, see the *ESXCLI Reference* documentation.

VMware SATPs

Storage Array Type Plug-ins (SATPs) are responsible for array-specific operations. The SATPs are submodules of the VMware NMP.

ESXi offers an SATP for every type of array that VMware supports. ESXi also provides default SATPs that support non-specific active-active, active-passive, ALUA, and local devices.

Each SATP accommodates special characteristics of a certain class of storage arrays. The SATP can perform the array-specific operations required to detect path state and to activate an inactive path. As a result, the NMP module itself can work with multiple storage arrays without having to be aware of the storage device specifics.

Generally, the NMP determines which SATP to use for a specific storage device and associates the SATP with the physical paths for that storage device. The SATP implements the tasks that include the following:

- Monitors the health of each physical path.
- Reports changes in the state of each physical path.
- Performs array-specific actions necessary for storage fail-over. For example, for active-passive devices, it can activate passive paths.

ESXi includes several generic SATP modules for storage arrays.

VMW_SATP_LOCAL

SATP for local direct-attached devices.

As of vSphere 6.5 Update 2 release, VMW_SATP_LOCAL provides multipathing support for the local devices, except the devices in 4K native format. To claim multiple paths to the local devices, you are no longer required to use other SATPs as you were in the earlier vSphere releases.

VMW_SATP_LOCAL supports the VMW_PSP_MRU and VMW_PSP_FIXED path selection plug-ins, but does not support VMW_PSP_RR.

VMW_SATP_DEFAULT_AA

Generic SATP for active-active arrays.

VMW_SATP_DEFAULT_AP

Generic SATP for active-passive arrays.

VMW_SATP_ALUA

SATP for ALUA-compliant arrays.

For more information, see the *VMware Compatibility Guide* and the *ESXCLI Reference* documentation.

Display SATPs for the Host

Use the `esxcli` command to list VMware NMP SATPs loaded into the system. Display information about the SATPs.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To list VMware SATPs, run the following command:

  `esxcli storage nmp satp list`

Results

For each SATP, the output displays information that shows the type of storage array or system the SATP supports. The output also shows the default PSP for any LUNs that use this SATP. `Placeholder (plugin not loaded)` in the Description column indicates that the SATP is not loaded.

Example: Displaying SATPs for the Host

```
# esxcli storage nmp satp list
Name                 Default PSP    Description
VMW_SATP_MSA         VMW_PSP_MRU    Placeholder (plugin not loaded)
VMW_SATP_ALUA        VMW_PSP_MRU    Placeholder (plugin not loaded)
VMW_SATP_DEFAULT_AP  VMW_PSP_MRU    Placeholder (plugin not loaded)
VMW_SATP_SVC         VMW_PSP_FIXED  Placeholder (plugin not loaded)
VMW_SATP_EQL         VMW_PSP_FIXED  Placeholder (plugin not loaded)
VMW_SATP_INV         VMW_PSP_FIXED  Placeholder (plugin not loaded)
VMW_SATP_EVA         VMW_PSP_FIXED  Placeholder (plugin not loaded)
VMW_SATP_ALUA_CX     VMW_PSP_RR     Placeholder (plugin not loaded)
VMW_SATP_SYMM        VMW_PSP_RR     Placeholder (plugin not loaded)
VMW_SATP_CX          VMW_PSP_MRU    Placeholder (plugin not loaded)
VMW_SATP_LSI         VMW_PSP_MRU    Placeholder (plugin not loaded)
VMW_SATP_DEFAULT_AA  VMW_PSP_FIXED  Supports non-specific active/active arrays
VMW_SATP_LOCAL       VMW_PSP_FIXED  Supports direct attached devices
```

For more information about the command, see the *ESXCLI Concepts and Examples* and *ESXCLI Reference* documentation.

VMware High Performance Plug-In and Path Selection Schemes

VMware provides the High-Performance Plug-in (HPP) to improve the performance of storage devices on your ESXi host.

The HPP replaces the NMP for high-speed devices, such as NVMe. The HPP is the default plug-in that claims NVMe-oF targets. ESXi supports end to end NVMe without emulation and SCSI to NVMe emulation. The HPP supports only active/active and implicit ALUA targets.

HPP Support	vSphere 8.0 Update 3
Storage devices	Local NVMe and SCSI Shared NVMe-oF (active/active and implicit ALUA targets only)
Multipathing	Yes
Second-level plug-ins	No
SCSI-3 persistent reservations	No
4Kn devices with software emulation	Yes

Path Selection Schemes

To support multipathing, the HPP uses the Path Selection Schemes (PSS) when selecting physical paths for I/O requests.

You can use the vSphere Client or the `esxcli` command to change the default path selection mechanism.

For information about configuring the path mechanisms in the vSphere Client, see Change the Path Selection Policy. To configure with the `esxcli` command, see ESXi esxcli HPP Commands.

ESXi supports the following path selection mechanisms.

FIXED

With this scheme, a designated preferred path is used for I/O requests. If the preferred path is not assigned, the host selects the first working path discovered at the boot time. If the preferred path becomes unavailable, the host selects an alternative available path. The host returns to the previously defined preferred path when it becomes available again.

When you configure **FIXED** as a path selection mechanism, select the preferred path.

LB-RR (Load Balance - Round Robin)

This is the default scheme for the devices claimed by HPP. After transferring a specified number of bytes or I/Os on a current path, the scheme selects the path using the round robin algorithm.

To configure the **LB-RR** path selection mechanism, specify the following properties:

- **IOPS** indicates the I/O count on the path to be used as criteria to switch a path for the device.
- **Bytes** indicates the byte count on the path to be used as criteria to switch a path for the device.

LB-IOPS (Load Balance - IOPs)

After transferring a specified number of I/Os on a current path, default is 1000, the system selects an optimal path that has the least number of outstanding I/Os.

When configuring this mechanism, specify the **IOPS** parameter to indicate the I/O count on the path to be used as criteria to switch a path for the device.

LB-BYTES (Load Balance - Bytes)

After transferring a specified number of bytes on a current path, default is 10 MB, the system selects an optimal path that has the least number of outstanding bytes.

To configure this mechanism, use the **Bytes** parameter to indicate the byte count on the path to be used as criteria to switch a path for the device.

Load Balance - Latency (LB-Latency)

To achieve better load balancing results, the mechanism dynamically selects an optimal path by considering the following path characteristics:

- **Latency evaluation time** parameter indicates at what time interval, in milliseconds, the latency of paths must be evaluated.
- **Sampling I/Os per path** parameter controls how many sample I/Os must be issued on each path to calculate latency of the path.

HPP Best Practices

To achieve the fastest throughput from a high-speed storage device, follow these recommendations.

- Use the vSphere version that supports the HPP.
- Use the HPP for local NVMe and SCSI devices, and NVMe-oF devices.
- If you use NVMe over Fibre Channel devices, follow general recommendations for Fibre Channel storage. See Chapter 3 Using ESXi with Fibre Channel SAN.
- If you use NVMe-oF, do not mix transport types to access the same namespace.
- When using NVMe-oF namespaces, make sure that active paths are presented to the host. The namespaces cannot be registered until the active path is discovered.
- When you configure your VMs, you can use VMware Paravirtual controllers or add NVMe controllers. Both types have their advantages and disadvantages. To check which works best for your environment, see SCSI, SATA, and NVMe Storage Controller Conditions, Limitations, and Compatibility in the *vSphere Virtual Machine Administration* documentation.
- Set the latency sensitive threshold.
- If a single VM drives a significant share of the device's I/O workload, consider spreading the I/O across multiple virtual disks. Attach the disks to separate virtual controllers in the VM.

  Otherwise, I/O throughput might be limited due to saturation of the CPU core responsible for processing I/Os on a particular virtual storage controller.

For information about device identifiers for NVMe devices that support only NGUID ID format, see NVMe Devices with NGUID Device Identifiers.

Enable the High-Performance Plug-In and the Path Selection Schemes

The high-performance plug-in (HPP) is the default plug-in that claims local NVMe and SCSI devices, and NVMe-oF targets. If necessary, you can replace it with NMP. In vSphere version 7.0 Update 1 and earlier, NMP remains the default plug-in for local NVMe and SCSI devices, but you can replace it with HPP.

Use the `esxcli storage core claimrule add` command to enable the HPP or NMP on your ESXi host.

To run the `esxcli storage core claimrule add`, you can use the ESXi Shell or vSphere CLI. For more information, see *Getting Started with ESXCLI* and *ESXCLI Reference*.

Examples in this topic demonstrate how to enable HPP and set up the path selection schemes (PSS).

Note Enabling the HPP is not supported on PXE booted ESXi hosts.

Prerequisites

Set up your VMware NVMe storage environment. For more information, see Chapter 8 About VMware NVMe Storage.

Procedure

1 Create an HPP claim rule by running the `esxcli storage core claimrule add` command.

Use one of the following methods to add the claim rule.

Method	Description
Based on the NVMe controller model	**`esxcli storage core claimrule add --type vendor --nvme-controller-model`** For example, **`esxcli storage core claimrule add --rule 429 --type vendor --nvme-controller-model "ABCD*" --plugin HPP`**
Based on the PCI vendor ID and subvendor ID	**`esxcli storage core claimrule add --type vendor --pci-vendor-id --pci-sub-vendor-id`** For example, **`esxcli storage core claimrule add --rule 429 --type vendor --pci-vendor-id 8086 --pci-sub-vendor-id 8086 --plugin HPP`**.

2 Configure the PSS.

Use one of the following methods.

Method	Description
Set the PSS based on the device ID	**esxcli storage hpp device set** For example, **esxcli storage hpp device set --device=*device* --pss=FIXED --path=preferred path**
Set the PSS based on the vendor/ model	Use the **--config-string** option with the **esxcli storage core claimrule add** command. For example, **esxcli storage core claimrule add -r 914 -t vendor -V *vendor* -M *model* -P HPP --config-string "pss=LB-Latency,latency-eval-time=40000"**

3 Reboot your host for your changes to take effect.

Set Latency Sensitive Threshold

When you use the HPP for your storage devices, set the latency sensitive threshold for the device, so that I/O can avoid the I/O scheduler.

By default, ESXi passes every I/O through the I/O scheduler. However, using the scheduler might create internal queuing, which is not efficient with the high-speed storage devices.

You can configure the latency sensitive threshold and enable the direct submission mechanism that helps I/O to bypass the scheduler. With this mechanism enabled, the I/O passes directly from PSA through the HPP to the device driver.

For the direct submission to work properly, the observed average I/O latency must be lower than the latency threshold you specify. If the I/O latency exceeds the latency threshold, the system stops the direct submission and temporarily reverts to using the I/O scheduler. The direct submission is resumed when the average I/O latency drops below the latency threshold again.

You can set the latency threshold for a family of devices claimed by HPP. Set the latency threshold using the vendor and model pair, the controller model, or PCIe vendor ID and sub vendor ID pair.

Procedure

1 Set the latency sensitive threshold for the device by running the following command:

esxcli storage core device latencythreshold set -t *value in milliseconds*

Use one of the following options.

Option	Example
Vendor/model	Set the latency sensitive threshold parameter for all devices with the indicated vendor and model: **esxcli storage core device latencythreshold set -v 'vendor1' -m 'model1' -t 10**
NVMe controller model	Set the latency sensitive threshold for all NVMe devices with the indicated controller model: **esxcli storage core device latencythreshold set -c 'controller_model1' -t 10**
PCIe vendor/subvendor ID	Set the latency sensitive threshold for devices with 0x8086 as PCIe vendor ID and 0x8086 as PCIe sub vendor ID. **esxcli storage core device latencythreshold set -p '8086' -s '8086' -t 10**

2 Verify that the latency threshold is set:

esxcli storage core device latencythreshold list

```
Device                Latency Sensitive Threshold
--------------------  ---------------------------
naa.55cd2e404c1728aa             0 milliseconds
naa.500056b34036cdfd             0 milliseconds
naa.55cd2e404c172bd6            50 milliseconds
```

3 Monitor the status of the latency sensitive threshold. Check VMkernel logs for the following entries:

- `Latency Sensitive Gatekeeper turned on for device` *device*`. Threshold of` *XX* `msec is larger than max completion time of` *YYY* `msec`
- `Latency Sensitive Gatekeeper turned off for device` *device*`. Threshold of` *XX* `msec is exceeded by command completed in` *YYY* `msec`

ESXi esxcli HPP Commands

You can use the ESXi Shell or vSphere CLI commands to configure and monitor the high-performance plug-in.

See *Getting Started with ESXCLI* for an introduction, and *ESXCLI Reference* for details of the `esxcli` command use.

Command	Description	Options
`esxcli storage hpp path list`	List the paths currently claimed by the high-performance plug-in.	`-d|--device=`*device* Display information for a specific device. `-p|--path=`*path* Limit the output to a specific path.
`esxcli storage hpp device list`	List the devices currently controlled by the high-performance plug-in.	`-d|--device=`*device* Show a specific device.

Command	Description	Options

`esxcli storage hpp device set`	Configure settings for an HPP device.	`-B|--bytes=long` Maximum bytes on the path, after which the path is switched.
		`--cfg-file` Update the configuration file and runtime with the new setting. If the device is claimed by another PSS, ignore any errors when applying to runtime configuration.
		`-d|--device=device` The HPP device upon which to operate. Use any of the UIDs that the device reports. Required.
		`-I|--iops=long` Maximum IOPS on the path, after which the path is switched.
		`-T|--latency-eval-time=long` Control at what interval, in ms, the latency of paths must be evaluated.
		`-L|--mark-device-local=bool` Set HPP to treat the device as local or not.
		`-M|--mark-device-ssd=bool` Specify whether or not the HPP treats the device as an SSD.
		`-p|--path=str` The path to set as the preferred path for the device.
		`-P|--pss=pss_name` The path selection scheme to assign to the device. If you do not specify the value, the system selects the default. For the description of path selection schemes, see VMware High Performance Plug-In and Path Selection Schemes. Options include:
		■ **FIXED**
		Use the `-p|--path=str` suboption to set the preferred path.
		■ **LB-Bytes**
		Use the `-B|--bytes=long` suboption to specify the input.
		■ **LB-IOPs**
		Use the `-I|--iops=long` suboption to specify the input.
		■ **LB-Latency**
		Suboptions include:
		`-T|--latency-eval-time=long`
		`-S|--sampling-ios-per-path=long`
		■ **LB-RR** Default
		Suboptions include:
		`-B|--bytes=long`
		`-I|--iops=long`

Command	Description	Options
		`-S|--sampling-ios-per-path=long` Control how many sample I/Os must be issued on each path to calculate latency of the path. `-U|--use-ano=bool` Set the option to `true` to include non-optimized paths in the set of active paths used to issue I/Os on this device. Otherwise, set the option to `false`.
`esxcli storage hpp device usermarkedssd list`	List the devices that were marked or unmarked as SSD by user.	`-d|--device=device` Limit the output to a specific device.

Viewing and Managing Storage Paths on ESXi Hosts

When you start your ESXi host or rescan your storage adapter, the host discovers all physical paths to storage devices available to the host. Based on a set of claim rules, the host determines which multipathing module, the NMP, HPP, or an MPP, owns the paths to a particular device.

The module that owns the device becomes responsible for managing the multipathing support for the device. By default, the host performs a periodic path evaluation every five minutes and assigns unclaimed paths to the appropriate module.

For the paths managed by the NMP module, a second set of claim rules is used. These rules assign an SATP and PSP modules to each storage device and determine which Storage Array Type Policy and Path Selection Policy to apply.

Use the vSphere Client to view the Storage Array Type Policy and Path Selection Policy assigned to a specific storage device. You can also check the status of all available paths for this storage device. If needed, you can change the default Path Selection Policy using the client.

To change the default multipathing module or SATP, modify claim rules using the vSphere CLI.

You can find some information about modifying claim rules in Using Claim Rules to Control ESXi Multipathing Modules.

To learn more about multipathing modules, see Multipathing Concepts and Acronyms.

View Storage Device Paths

View which multipathing policies the host uses for a specific storage device and the status of all available paths for this storage device.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices**.

4 Select the storage device whose paths you want to view.

5 Click the **Properties** tab and review the module that owns the device, for example, NMP or HPP.

Under Multipathing Policies, you can also see the Path Selection Policy and, if applicable, the Storage Array Type Policy assigned to the device.

6 Click the **Paths** tab to review all paths available for the storage device and the status of each path. The following path status information can appear:

Status	Description
Active (I/O)	Working path or multiple paths that currently transfer data.
Standby	Paths that are inactive. If the active path fails, they can become operational and start transferring I/O.
Disabled	Paths that are deactivated by the administrator.
Dead	Paths that are no longer available for processing I/O. A physical medium failure or array misconfiguration can cause this status.

If you are using the **Fixed** path policy, you can see which path is the preferred path. The preferred path is marked with an asterisk (*) in the Preferred column.

View Datastore Paths

Review the paths that connect to storage devices backing your VMFS datastores.

Procedure

1 In the vSphere Client, navigate to the datastore.

2 Click the **Configure** tab.

3 Click **Connectivity and Multipathing**.

4 Select a host to view multipathing details for its devices.

5 Under Multipathing Policies, review the module that owns the device, such as NMP. You can also see the Path Selection Policy and Storage Array Type Policy assigned to the device.

For example, you might see the following:

Path Selection Policy	Preferred Path
Storage Array Type Policy	VMW_SATP_LOCAL
Owner Plugin	NMP

6 Under Paths, review the device paths and the status of each path. The following path status information can appear:

Status	Description
Active (I/O)	Working path or multiple paths that currently transfer data.
Standby	Paths that are inactive. If the active path fails, they can become operational and start transferring I/O.

Status	Description
Disabled	Paths that are deactivated by the administrator.
Dead	Paths that are no longer available for processing I/O. A physical medium failure or array misconfiguration can cause this status.

If you are using the **Fixed** path policy, you can see which path is the preferred path. The preferred path is marked with an asterisk (*) in the Preferred column.

Change the Path Selection Policy

Generally, you do not need to change the default multipathing settings that your ESXi host uses for a specific storage device. If you want to make any changes, you can use the **Edit Multipathing Policies** dialog box to modify the path selection policy. You can also use this dialog box to change multipathing for SCSI-based protocol endpoints.

Procedure

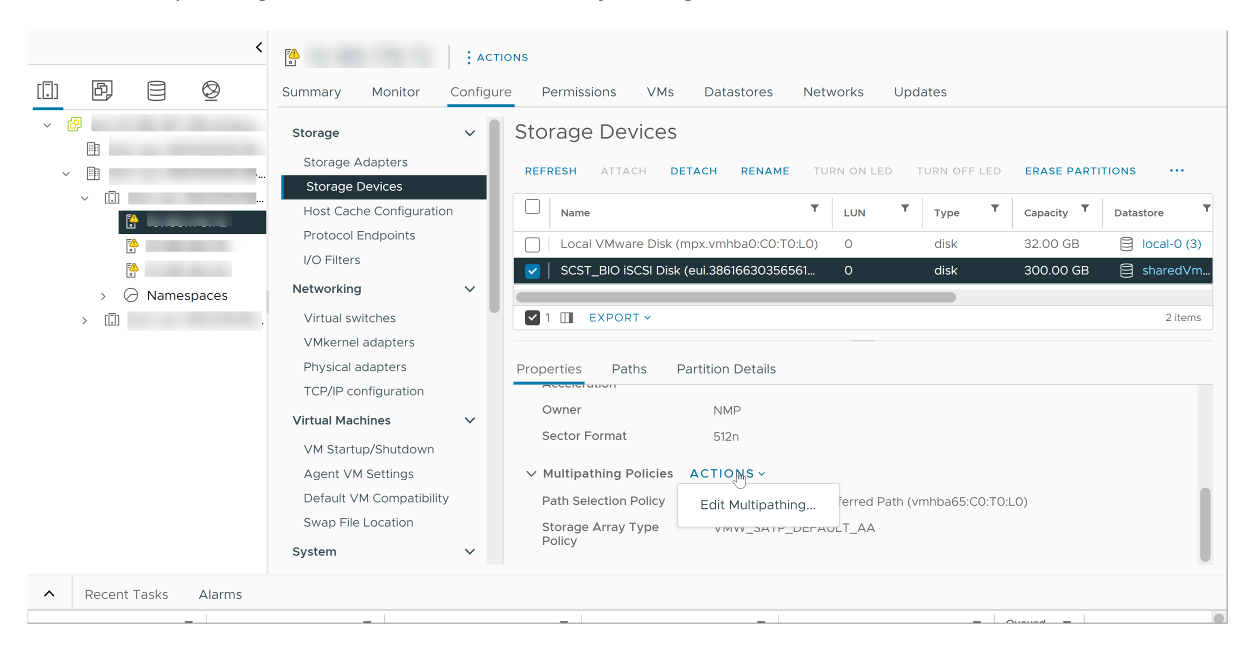
1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Devices** or **Protocol Endpoints**.

4 Select the item whose paths you want to change and click the **Properties** tab.

5 Under Multipathing Policies, select **Edit Multipathing** from the **Actions** menu.



6 Select a path policy and configure its settings. Your options change depending on the type of a storage device you use.

- For information about path policies for SCSI devices, see VMware Path Selection Plug-Ins and Policies.

- For information about path mechanisms for NVMe devices, see VMware High Performance Plug-In and Path Selection Schemes.

7 To save your settings and exit the dialog box, click **OK**.

Change Default Parameters for Latency Round Robin

On the ESXi host, you can activate the latency mechanism for the Round Robin path selection policy. The mechanism considers I/O bandwidth and path latency to select an optimal path for I/O. When using the latency mechanism, the Round Robin policy can dynamically select the optimal path and achieve better load balancing results.

When latency is activated, usage of paths for the I/Os depends on the workload. When a large number of paths is available, but the workload is not high enough, the system might not use some paths for I/Os.

You use the `esxcli` command to change the default parameters of the latency mechanism or deactivate the mechanism.

Prerequisites

Set the path selection policy to Round Robin. See Change the Path Selection Policy.

Procedure

1 Configure the latency mechanism by using the following command.

```
esxcli storage nmp psp roundrobin deviceconfig set --type=latency --device=device ID
```

The command takes the following parameters:

Parameter	Description
-S|--num-sampling-cycles=*sampling value*	When `--type` is set to `latency`, this parameter controls how many I/Os to use to calculate the average latency of each path. The default value of this parameter is 16.
-T|--latency-eval-time=*time in ms*	When `--type` is set to `latency`, this parameter controls the frequency at which the latency of paths is updated. Default is 3 minutes.

2 Verify whether the latency Round Robin and its parameters are configured correctly.

```
esxcli storage nmp psp roundrobin deviceconfig get --device=device ID
```

or

```
esxcli storage nmp device list --device=device ID
```

The following sample output shows the path's configuration:

```
 Path Selection Policy: VMW_PSP_RR
   Path Selection Policy Device Config:
{policy=latency,latencyEvalTime=180000,samplingCycles=16,curSamplingCycle=16,useANO=0;
CurrentPath=vmhba1:C0:T0:L0: NumIOsPending=0,latency=0}
```

What to do next

To deactivate the latency mechanism, in the Advanced System Settings for your host, change the `Misc.EnablePSPLatencyPolicy` parameter to 0.

Deactivate Storage Paths

You can temporarily deactivate paths for maintenance or other reasons.

You deactivate a path using the Paths panel. You have several ways to access the Paths panel, from a datastore, a storage device, an adapter, or a Virtual Volumes Protocol Endpoint view.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click one of the following items:

   - **Storage Adapters**
   - **Storage Devices**
   - **Protocol Endpoints**

4 In the right pane, select the item whose paths you want to deactivate, an adapter, storage device, or Protocol Endpoint, and click the **Paths** tab.

5 Select the path to deactivate and click **Disable**.

   The path's status changes to Disabled.

Using Claim Rules to Control ESXi Multipathing Modules

Claim rules determine which multipathing module owns the paths to a particular storage device. They also define the type of multipathing support that the host provides to the device.

To learn more about multipathing modules, see Multipathing Concepts and Acronyms.

The claim rules are listed in the host’s `/etc/vmware/esx.conf` file.

The rules fall into these categories:

- Core Claim Rules . These claim rules determine which multipathing module, the NMP, HPP, or a third-party MPP, claims the specific device.

- SATP Claim Rules. Depending on the device type, these rules assign a particular SATP submodule that provides vendor-specific multipathing management to the device.

You can use the `esxcli` commands to add or change the core and SATP claim rules. Typically, you add the claim rules to load a third-party MPP or to hide a LUN from your host. Changing claim rules might be necessary when default settings for a specific device are not sufficient.

For more information about commands available to manage PSA claim rules, see the *Getting Started with ESXCLI*.

For a list of storage arrays and corresponding SATPs and PSPs, see the Storage/SAN section of the *vSphere Compatibility Guide*.

Multipathing Considerations

Specific considerations apply when you manage storage multipathing plug-ins and claim rules.

The following considerations help you with multipathing:

- If no SATP is assigned to the device by the claim rules, the default SATP for iSCSI or FC devices is VMW_SATP_DEFAULT_AA. The default PSP is VMW_PSP_FIXED.
- When the system searches the SATP rules to locate a SATP for a given device, it searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules are searched. If no match occurs, NMP selects a default SATP for the device.
- If VMW_SATP_ALUA is assigned to a specific storage device, but the device is not ALUA-aware, no claim rule match occurs for this device. The device is claimed by the default SATP based on the device's transport type.
- The default PSP for all devices claimed by VMW_SATP_ALUA is VMW_PSP_MRU. The VMW_PSP_MRU selects an active/optimized path as reported by the VMW_SATP_ALUA, or an active/unoptimized path if there is no active/optimized path. This path is used until a better path is available (MRU). For example, if the VMW_PSP_MRU is currently using an active/unoptimized path and an active/optimized path becomes available, the VMW_PSP_MRU will switch the current path to the active/optimized one.
- While VMW_PSP_MRU is typically selected for ALUA arrays by default, certain ALUA storage arrays need to use VMW_PSP_FIXED. To check whether your storage array requires VMW_PSP_FIXED, see the *VMware Compatibility Guide* or contact your storage vendor. When using VMW_PSP_FIXED with ALUA arrays, unless you explicitly specify a preferred path, the ESXi host selects the most optimal working path and designates it as the default preferred path. If the host selected path becomes unavailable, the host selects an alternative available path. However, if you explicitly designate the preferred path, it will remain preferred no matter what its status is.
- By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

List Multipathing Claim Rules for the Host

Use the `esxcli` command to list available multipathing claim rules.

Claim rules indicate whether the NMP, HPP, or a third-party MPP manages a given physical path. Each claim rule identifies a set of paths based on the following parameters:

- Vendor/model strings
- Transportation, such as SATA, IDE, Fibre Channel
- Adapter, target, or LUN location
- Device driver, for example, Mega-RAID

Procedure

- List the multipathing claim rules by running the **`esxcli storage core claimrule list --claimrule-class=MP`** command.

  If you do not use the `claimrule-class` option, the MP rule class is implied.

Example: Sample Output of the esxcli storage core claimrule list Command

```
Rule Class  Rule   Class    Type       Plugin     Matches
MP            10   runtime  vendor     HPP        vendor=NVMe model=*
MP            10   file     vendor     HPP        vendor=NVMe model=*
MP            50   runtime  transport  NMP        transport=usb
MP            51   runtime  transport  NMP        transport=sata
MP            52   runtime  transport  NMP        transport=ide
MP            53   runtime  transport  NMP        transport=block
MP            54   runtime  transport  NMP        transport=unknown
MP           101   runtime  vendor     MASK_PATH  vendor=DELL model=Universal Xport
MP           101   file     vendor     MASK_PATH  vendor=DELL model=Universal Xport
MP           200   runtime  vendor     MPP_1      vendor=NewVend model=*
MP           200   file     vendor     MPP_1      vendor=NewVend model=*
MP           201   runtime  location   MPP_2      adapter=vmhba41 channel=* target=* lun=*
MP           201   file     location   MPP_2      adapter=vmhba41 channel=* target=* lun=*
MP           202   runtime  driver     MPP_3      driver=megaraid
MP           202   file     driver     MPP_3      driver=megaraid
MP           65535 runtime  vendor     NMP        vendor=* model=*
```

This example indicates the following:

- The NMP claims all paths connected to storage devices that use the USB, SATA, IDE, and Block SCSI transportation.
- The rules for HPP, MPP_1, MPP_2, and MPP_3 have been added, so that the modules can claim specified devices. For example, the HPP claims all devices with vendor NVMe. All devices handled by the inbox nvme driver are claimed regardless of the actual vendor. The MPP_1 module claims all paths connected to any model of the NewVend storage array.

- You can use the MASK_PATH module to hide unused devices from your host. By default, the PSA claim rule 101 masks Dell array pseudo devices with a vendor string `DELL` and a model string `Universal Xport`.
- The Rule Class column in the output describes the category of a claim rule. It can be MP (multipathing plug-in), Filter, or VAAI.
- The Class column shows which rules are defined and which are loaded. The `file` parameter in the Class column indicates that the rule is defined. The `runtime` parameter indicates that the rule has been loaded into your system. For a user-defined claim rule to be active, two lines with the same rule number must exist, one line for the rule with the `file` parameter and another line with `runtime`. Several default system-defined claim rules have only one line with the Class of `runtime`. You cannot modify these rules.
- The default rule 65535 assigns all unclaimed paths to the NMP. Do not delete this rule.

Add Multipathing Claim Rules

Use the `esxcli` commands to add a multipathing PSA claim rule to the set of claim rules on the system. For the new claim rule to be active, you first define the rule and then load it into your system.

Examples when you add a PSA claim rule include the following:

- You load a new third-party MPP and must define the paths that this module claims.
- You must enable the native HPP.

Warning You cannot create rules where two different plug-ins claim paths to the same device. Your attempts to create these claim rules fail with a warning in `vmkernel.log`.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 To define a new claim rule, use the following command:

   `esxcli storage core claimrule add`

   The command takes the following options:

Option	Description
`-A|--adapter=<adapter>`	Adapter of the paths to use. Valid only if `--type` is `location`.
`-u|--autoassign`	Adds a claim rule based on its characteristics. The rule number is not required.
`-C|--channel=<channel>`	Channel of the paths to use. Valid only if `--type` is `location`.

Option	Description
-c|--claimrule-class=<cl>	Claim rule class to use in this operation. You can specify MP (default), Filter, or VAAI. To configure hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plug-in. See Add Hardware Acceleration Claim Rules for detailed instructions.
-d|--device=<device_uid>	UID of the device. Valid only when --type is device.
-D|--driver=<driver>	Driver for the HBA of the paths to use. Valid only if --type is driver.
-f|--force	Force claim rules to ignore validity checks and install the rule anyway.
--force-reserved	Override protection of reserved rule ID ranges. Reserved claim rules are the rules with an ID below 100. You can use them to reassign local devices to specific plug-ins, for example, the NVMe device to HPP.
--if-unset=<str>	Run this command if this advanced user variable is not set to 1.
-i|--iqn=<iscsi_name>	iSCSI Qualified Name for the target. Valid only when --type is target.
-L|--lun=<lun_id>	LUN of the paths. Valid only if --type is location. LUN ID must not be higher than the value of the advanced configuration option /Disk/MaxLUN.
-M|--model=<model>	Model of the paths to use. Valid only if --type is vendor. Valid values are values of the Model string from the SCSI inquiry string. Run vicfg-scsidevs <conn_options> -l on each device to see model string values.
-P|--plugin=<plugin>	PSA plug-in to use. The values are NMP, MASK_PATH, or HPP. Third parties can also provide their own PSA plug-ins. Required.
-r|--rule=<rule_ID>	Rule ID to use. The rule ID indicates the order in which the claim rule is to be evaluated. User-defined claim rules are evaluated in numeric order starting with 101. You can run esxcli storage core claimrule list to determine which rule IDs are available.
-T|--target=<target>	Target of the paths to use. Valid only if --type is location.
-R|--transport=<transport>	Transport of the paths to use. Valid only if --type is transport. The following values are supported. ■ block — block storage ■ fc — Fibre Channel ■ iscsivendor — iSCSI ■ iscsi — not currently used ■ ide — IDE storage ■ sas — SAS storage ■ sata — SATA storage ■ usb — USB storage ■ parallel — parallel ■ fcoe — FCoE ■ unknown

Option	Description
-t|--type=<type>	Type of matching to use for the operation. Valid values are the following. Required. ■ `vendor` ■ `location` ■ `driver` ■ `transport` ■ `device` ■ `target`
-V|--vendor=<vendor>	Vendor of the paths to use. Valid only if `--type` is `vendor`. Valid values are values of the vendor string from the SCSI inquiry string. Run `vicfg-scsidevs <conn_options> -l` on each device to see vendor string values.
--wwnn=<wwnn>	World-Wide Node Number for the target.
--wwpn=<wwpn>	World-Wide Port Number for the target.
-a|--xcopy-use-array-values	Use the array reported values to construct the XCOPY command to be sent to the storage array. This applies to VAAI claim rules only.
-s|--xcopy-use-multi-segs	Use multiple segments when issuing an XCOPY request. Valid only if `--xcopy-use-array-values` is specified.
-m|--xcopy-max-transfer-size	Maximum data transfer size in MB when you use a transfer size different than array reported. Valid only if `--xcopy-use-array-values` is specified.
-k|--xcopy-max-transfer-size-kib	Maximum transfer size in KiB for the XCOPY commands when you use a transfer size different than array reported. Valid only if `--xcopy-use-array-values` is specified.

2 To load the new claim rule into your system, use the following command:

esxcli storage core claimrule load

This command loads all newly created multipathing claim rules from the `esx.conf` configuration file into the VMkernel. The command has no options.

3 To apply claim rules that are loaded, use the following command:

esxcli storage core claimrule run

The command takes the following options:

Option	Description
-A|--adapter=<adapter>	If `--type` is `location`, name of the HBA for the paths to run the claim rules on. To run claim rules on paths from all adapters, omit this option.
-C|--channel=<channel>	If `--type` is `location`, indicate the channel of the paths to use in this operation. To run claim rules on paths with any channel number, omit this option.
-c|--claimrule-class=<cl>	Claim rule class to use in this operation.
-d|--device=<device_uid>	UID of the device.

Option	Description
-L|--lun=<lun_id>	If `--type` is `location`, indicate the LUN of the paths to run claim rules on. To run claim rules on paths with any LUN, omit this option.
-p|--path=<path_uid>	If `--type` is `path`, this option indicates the unique path identifier (UID) or the runtime name of a path to run claim rules on.
-T|--target=<target>	If `--type` is `location`, indicate the target of the paths to run claim rules on. To run claim rules on paths with any target number, omit this option.
-t|--type=<location|path|all>	Type of claim to perform. By default, uses `all`, which means claim rules run without restriction to specific paths. Valid values are `location`, `path`, and `all`.
-w|--wait	You can use this option only if you also use `--type all`. If the option is included, the claim waits for paths to settle before running the claim operation. In that case, the system does not start the claiming process until it is likely that all paths on the system have appeared before starting the claim process. After the claiming process has started, the command does not return until device registration has completed. If you add or remove paths during the claiming or the discovery process, this option might not work correctly.

Example: Defining Multipathing Claim Rules

In the following example, you add and load rule # 500. The rule claims all paths with the NewMod model string and the NewVend vendor string for the NMP plug-in.

esxcli storage core claimrule add -r 500 -t vendor -V NewVend -M NewMod -P NMP

esxcli storage core claimrule load

After you run the **esxcli storage core claimrule list** command, you can see the new claim rule appearing on the list.

The following output indicates that the claim rule 500 has been loaded into the system and is active.

```
Rule Class  Rule   Class    Type      Plugin    Matches
...         ...    ...      ...       ...       ...
MP          500    runtime  vendor    NMP       vendor=NewVend model=NewMod
MP          500    file     vendor    NMP       vendor=NewVend model=NewMod
```

Delete Multipathing Claim Rules

Use the `esxcli` commands to remove a multipathing PSA claim rule from the set of claim rules on the system.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Delete a claim rule from the set of claim rules.

esxcli storage core claimrule remove

Note By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

The command takes the following options:

Option	Description
`-c|--claimrule-class=<str>`	Indicate the claim rule class (MP, Filter, VAAI).
`-P|--plugin=<str>`	Indicate the plug-in.
`-r|--rule=<long>`	Indicate the rule ID.

This step removes the claim rule from the File class.

2 Remove the claim rule from the system.

esxcli storage core claimrule load

This step removes the claim rule from the Runtime class.

Mask Paths

You can prevent the host from accessing storage devices or LUNs or from using individual paths to a LUN. Use the `esxcli` commands to mask the paths. When you mask paths, you create claim rules that assign the MASK_PATH plug-in to the specified paths.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Check what the next available rule ID is.

esxcli storage core claimrule list

The claim rules that you use to mask paths have rule IDs in the range from 101 through 200. If this command shows that rules 101 and 102 exist, you can specify 103 for the rule to add.

2 Assign the MASK_PATH plug-in to a path by creating a new claim rule for the plug-in.

esxcli storage core claimrule add -P MASK_PATH

3 Load the MASK_PATH claim rule into your system.

esxcli storage core claimrule load

4 Verify that the MASK_PATH claim rule was added correctly.

esxcli storage core claimrule list

5 If a claim rule for the masked path exists, remove the rule.

esxcli storage core claiming unclaim

6 Run the path claiming rules.

esxcli storage core claimrule run

Results

After you assign the MASK_PATH plug-in to a path, the path state becomes irrelevant and is no longer maintained by the host. As a result, commands that display the masked path's information might show the path state as dead.

Example: Masking a LUN

In this example, you mask the LUN 20 on targets T1 and T2 accessed through storage adapters vmhba2 and vmhba3.

1
```
#esxcli storage core claimrule list
```

2
```
#esxcli storage core claimrule add -P MASK_PATH -r 109 -t location -A vmhba2 -C 0 -T 1 -L
20
#esxcli storage core claimrule add -P MASK_PATH -r 110 -t location -A vmhba3 -C 0 -T 1 -L
20
#esxcli storage core claimrule add -P MASK_PATH -r 111 -t location -A vmhba2 -C 0 -T 2 -L
20
#esxcli storage core claimrule add -P MASK_PATH -r 112 -t location -A vmhba3 -C 0 -T 2 -L
20
```

3
```
#esxcli storage core claimrule load
```

4
```
#esxcli storage core claimrule list
```

5
```
#esxcli storage core claiming unclaim -t location -A vmhba2
#esxcli storage core claiming unclaim -t location -A vmhba3
```

6
```
#esxcli storage core claimrule run
```

Unmask Paths

When you need the host to access the masked storage device, unmask the paths to the device.

Note When you run an unclaim operation using a device property, for example, device ID or vendor, the paths claimed by the MASK_PATH plug-in are not unclaimed. The MASK_PATH plug-in does not track any device property of the paths that it claims.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Delete the MASK_PATH claim rule.

esxcli storage core claimrule remove -r rule#

2 Verify that the claim rule was deleted correctly.

esxcli storage core claimrule list

3 Reload the path claiming rules from the configuration file into the VMkernel.

esxcli storage core claimrule load

4 Run the **esxcli storage core claiming unclaim** command for each path to the masked storage device.

For example:

esxcli storage core claiming unclaim -t location -A vmhba0 -C 0 -T 0 -L 149

5 Run the path claiming rules.

esxcli storage core claimrule run

Results

Your host can now access the previously masked storage device.

Define NMP SATP Rules

The NMP SATP claim rules define which SATP manages a storage device. Usually, you can use the default SATPs provided for storage devices. If default settings are not sufficient, use the `esxcli` commands to change the SATP for a specific device.

You might need to create an SATP rule when you install a third-party SATP for a specific storage array.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 To add a claim rule for a specific SATP, run the **esxcli storage nmp satp rule add** command. The command takes the following options.

Option	Description
-b|--boot	This rule is a system default rule added at boot time. Do not modify `esx.conf` or add to a host profile.
-c|--claim-option=*string*	Set the claim option string when adding a SATP claim rule.
-e|--description=*string*	Set the claim rule description when adding a SATP claim rule.

Option	Description
-d|--device=*string*	Set the device when adding SATP claim rules. Device rules are mutually exclusive with vendor/model and driver rules.
-D|--driver=*string*	Set the driver string when adding a SATP claim rule. Driver rules are mutually exclusive with vendor/model rules.
-f|--force	Force claim rules to ignore validity checks and install the rule anyway.
-h|--help	Show the help message.
-M|--model=*string*	Set the model string when adding SATP a claim rule. Vendor/Model rules are mutually exclusive with driver rules.
-o|--option=*string*	Set the option string when adding a SATP claim rule.
-P|--psp=*string*	Set the default PSP for the SATP claim rule.
-O|--psp-option=*string*	Set the PSP options for the SATP claim rule.
-s|--satp=*string*	The SATP for which a new rule is added.
-R|--transport=*string*	Set the claim transport type string when adding a SATP claim rule.
-t|--type=*string*	Set the claim type when adding a SATP claim rule.
-V|--vendor=*string*	Set the vendor string when adding SATP claim rules. Vendor/Model rules are mutually exclusive with driver rules.

Note When searching the SATP rules to locate a SATP for a given device, the NMP searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules. If there is still no match, NMP selects a default SATP for the device.

2 Reboot your host.

Example: Defining an NMP SATP Rule

The following sample command assigns the VMW_SATP_INV plug-in to manage storage arrays with vendor string NewVend and model string NewMod.

```
# esxcli storage nmp satp rule add -V NewVend -M NewMod -s VMW_SATP_INV
```

When you run the **esxcli storage nmp satp list -s VMW_SATP_INV** command, you can see the new rule on the list of VMW_SATP_INV rules.

Scheduling Queues for Virtual Machine I/Os in vSphere Environment

By default, vSphere provides a mechanism that creates scheduling queues for every virtual machine file. Each file, for example `.vmdk`, gets its own bandwidth controls. You can use the vSphere Client or the `esxcli` commands to turn off the mechanism.

This mechanism ensures that I/O for a particular virtual machine file goes into its own separate queue and avoids interfering with I/Os from other files.

Edit Per File I/O Scheduling in the vSphere Client

The advanced `VMkernel.Boot.isPerFileSchedModelActive` parameter controls the per file I/O scheduling mechanism on VMFS and NFS 3 datastores. On the ESXi host, the mechanism is turned on by default. You can turn off the mechanism using the **Advanced System Settings** dialog box.

If you turn off the per file I/O scheduling model, your host reverts to a legacy scheduling mechanism. The legacy scheduling maintains only one I/O queue for each virtual machine and storage device pair. All I/Os between the virtual machine and its virtual disks are moved into this queue. As a result, I/Os from different virtual disks might interfere with each other in sharing the bandwidth and affect each other's performance.

Note Do not deactivate per file scheduling if you have the HPP plug-in and the latency sensitive threshold parameter configured for high-speed local devices. Turning off per file scheduling might cause unpredictable behavior.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 Edit the value of the **VMkernel.Boot.isPerFileSchedModelActive** parameter.

Option	Description
True	Turn on the per file scheduling mechanism. By default, the parameter is activated.
False	Deactivate the per file scheduling mechanism.

5 Reboot the host for the changes to take effect.

Use esxcli Commands to Edit Per File I/O Scheduling

You can use the `esxcli` commands to change the I/O scheduling capability for VMFS, NFS 3, and NFS 4.1 datastores on your ESXi host. The capability is turned on by default.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- To set or edit per file I/O scheduling, run the following commands:

Option	Description
esxcli system settings kernel set -s isPerFileSchedModelActive -v FALSE	Turn off per file I/O scheduling for VMFS and NFS 3.
esxcli system settings kernel set -s isPerFileSchedModelActive -v TRUE	Turn on per file I/O scheduling for VMFS and NFS 3.
esxcli system module parameters list -m nfs41client	Display the current status of NFS 4.1 file based scheduler.
esxcli system module parameters set -m nfs41client -p fileBasedScheduler=0	Turn off file based scheduler for NFS 4.1.
esxcli system module parameters set -m nfs41client -p fileBasedScheduler=1	Turn on file based scheduler for NFS 4.1.

Raw Device Mapping in vSphere

11

Raw device mapping (RDM) in vSphere provides a mechanism for a virtual machine to have direct access to a LUN on the physical storage subsystem.

The following topics contain information about RDMs in vSphere and provide instructions on how to create and manage RDMs.

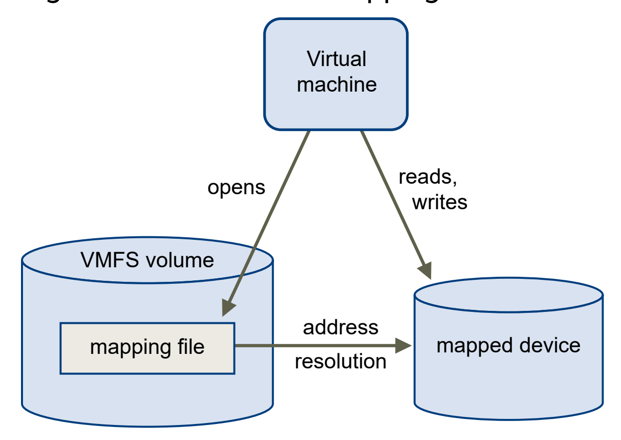
Read the following topics next:

Introduction to Raw Device Mapping in vSphere

An RDM is a mapping file in a separate vSphere VMFS volume that acts as a proxy for a raw physical storage device. With RDM, a virtual machine can access and use the storage device directly. The RDM contains metadata for managing and redirecting disk access to the physical device.

The file gives you some of the advantages of direct access to a physical device, but keeps some advantages of a virtual disk in VMFS. As a result, it merges the VMFS manageability with the raw device access.

Figure 11-1. Raw Device Mapping



Typically, you use VMFS datastores for most virtual disk storage. On certain occasions, you might use raw LUNs or logical disks located in a SAN.

For example, you might use raw LUNs with RDMs in the following situations:

- When SAN snapshot or other layered applications run in the virtual machine. The RDM enables backup offloading systems by using features inherent to the SAN.
- In any MSCS clustering scenario that spans physical hosts, such as virtual-to-virtual clusters and physical-to-virtual clusters. In this case, cluster data and quorum disks are configured as RDMs rather than as virtual disks on a shared VMFS.

Think of an RDM as a symbolic link from a VMFS volume to a raw LUN. The mapping makes LUNs appear as files in a VMFS volume. The RDM, not the raw LUN, is referenced in the virtual machine configuration. The RDM contains a reference to the raw LUN.

Two compatibility modes are available for RDMs:

- In the virtual compatibility mode, the RDM acts like a virtual disk file. The RDM can use snapshots.
- In the physical compatibility mode, the RDM offers direct access to the SCSI device for those applications that require lower-level control.

Benefits of Raw Device Mapping

An RDM provides a number of benefits, but it should not be used in every situation. In general, virtual disk files are preferable to RDMs for manageability. However, when you need raw devices, you must use the RDM.

RDM offers several benefits.

User-Friendly Persistent Names

Provides a user-friendly name for a mapped device. When you use an RDM, you do not need to refer to the device by its device name. You refer to it by the name of the mapping file, for example:

```
/vmfs/volumes/myVolume/myVMDirectory/myRawDisk.vmdk
```

Dynamic Name Resolution

Stores unique identification information for each mapped device. VMFS associates each RDM with its current SCSI device, regardless of changes in the physical configuration of the server because of adapter hardware changes, path changes, device relocation, and so on.

Distributed File Locking

Makes it possible to use VMFS distributed locking for raw SCSI devices. Distributed locking on an RDM makes it safe to use a shared raw LUN without losing data when two virtual machines on different servers try to access the same LUN.

File Permissions

Makes file permissions possible. The permissions of the mapping file are enforced at file-open time to protect the mapped volume.

File System Operations

Makes it possible to use file system utilities to work with a mapped volume, using the mapping file as a proxy. Most operations that are valid for an ordinary file can be applied to the mapping file and are redirected to operate on the mapped device.

Snapshots

Makes it possible to use virtual machine snapshots on a mapped volume. Snapshots are not available when the RDM is used in physical compatibility mode.

vMotion

Lets you migrate a virtual machine with vMotion. The mapping file acts as a proxy to allow vCenter Server to migrate the virtual machine by using the same mechanism that exists for migrating virtual disk files.

Figure 11-2. vMotion of a Virtual Machine Using Raw Device Mapping

SAN Management Agents

Makes it possible to run some SAN management agents inside a virtual machine. Similarly, any software that needs to access a device by using hardware-specific SCSI commands can

be run in a virtual machine. This kind of software is called SCSI target-based software. When you use SAN management agents, select a physical compatibility mode for the RDM.

N-Port ID Virtualization (NPIV)

Makes it possible to use the NPIV technology that allows a single Fibre Channel HBA port to register with the Fibre Channel fabric using several worldwide port names (WWPNs). This ability makes the HBA port appear as multiple virtual ports, each having its own ID and virtual port name. Virtual machines can then claim each of these virtual ports and use them for all RDM traffic.

Note You can use NPIV only for virtual machines with RDM disks.

VMware works with vendors of storage management software to ensure that their software functions correctly in environments that include ESXi. Some applications of this kind are:

- SAN management software
- Storage resource management (SRM) software
- Snapshot software
- Replication software

Such software uses a physical compatibility mode for RDMs so that the software can access SCSI devices directly.

Various management products are best run centrally (not on the ESXi machine), while others run well on the virtual machines. VMware does not certify these applications or provide a compatibility matrix. To find out whether a SAN management application is supported in an ESXi environment, contact the SAN management software provider.

RDM Considerations and Limitations

Certain considerations and limitations exist when you use RDMs.

- The RDM is not available for direct-attached block devices or certain RAID devices. The RDM uses a SCSI serial number to identify the mapped device. Because block devices and some direct-attach RAID devices do not export serial numbers, they cannot be used with RDMs.
- If you are using the RDM in physical compatibility mode, you cannot use a snapshot with the disk. Physical compatibility mode allows the virtual machine to manage its own, storage-based, snapshot or mirroring operations.

  Virtual machine snapshots are available for RDMs with virtual compatibility mode.
- You cannot map to a disk partition. RDMs require the mapped device to be a whole LUN.
- If you use vMotion to migrate virtual machines with RDMs, make sure to maintain consistent LUN IDs for RDMs across all participating ESXi hosts.

Characteristics of Raw Device Mapping in vSphere

An RDM is a special mapping file in a vSphere VMFS volume that manages metadata for its mapped device. The mapping file is presented to the management software as an ordinary disk file, available for the usual file-system operations. To the virtual machine, the storage virtualization layer presents the mapped device as a virtual SCSI device.

Key contents of the metadata in the mapping file include the location of the mapped device (name resolution), the locking state of the mapped device, permissions, and so on.

RDM Virtual and Physical Compatibility Modes

You can use RDMs in virtual compatibility or physical compatibility modes. Virtual mode specifies full virtualization of the mapped device. Physical mode specifies minimal SCSI virtualization of the mapped device, allowing the greatest flexibility for SAN management software.

In virtual mode, the VMkernel sends only READ and WRITE to the mapped device. The mapped device appears to the guest operating system exactly the same as a virtual disk file in a VMFS volume. The real hardware characteristics are hidden. If you are using a raw disk in virtual mode, you can realize the benefits of VMFS such as advanced file locking for data protection and snapshots for streamlining development processes. Virtual mode is also more portable across storage hardware than physical mode, presenting the same behavior as a virtual disk file.

In physical mode, the VMkernel passes all SCSI commands to the device, with one exception: the REPORT LUNs command is virtualized so that the VMkernel can isolate the LUN to the owning virtual machine. Otherwise, all physical characteristics of the underlying hardware are exposed. Physical mode is useful to run SAN management agents or other SCSI target-based software in the virtual machine. Physical mode also allows virtual-to-physical clustering for cost-effective high availability.

VMFS5 and VMFS6 support greater than 2 TB disk size for RDMs in virtual and physical modes.

Dynamic Name Resolution

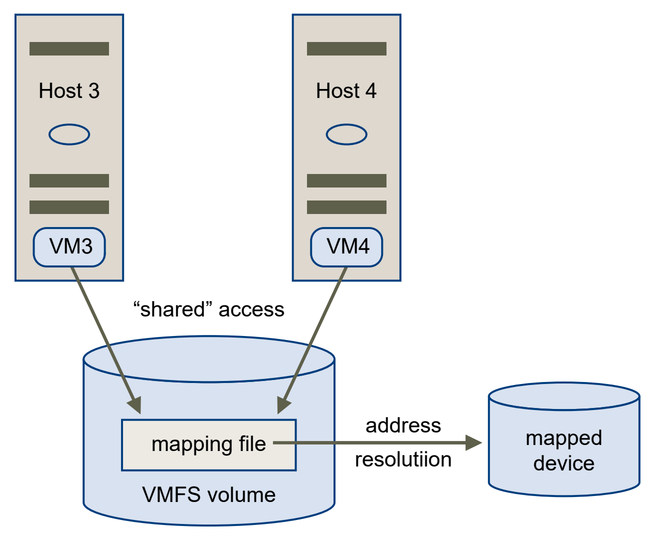
The RDM file supports dynamic name resolution when a path to a raw device changes.

VMFS uniquely identifies all mapped storage devices, and the identification is stored in its internal data structures. Any change in the path to a raw device, such as a Fibre Channel switch failure or the addition of a new HBA, can change the device name. Dynamic name resolution resolves these changes and automatically associates the original device with its new name.

Raw Device Mapping with Virtual Machine Clusters

Use an RDM with virtual machine clusters that require access to the same raw LUN for failover scenarios. The setup is similar to that of a virtual machine cluster that accesses the same virtual disk file, but an RDM replaces the virtual disk file.

Figure 11-3. Access from Clustered Virtual Machines



Comparing Available SCSI Device Access Modes

The ways of accessing a SCSI-based storage device include a virtual disk file on a VMFS datastore, virtual mode RDM, and physical mode RDM.

The following table provides a comparison of features available with the different modes.

Table 11-1. Features Available with Virtual Disks and Raw Device Mappings

ESXi Features	Virtual Disk File	Virtual Mode RDM	Physical Mode RDM
SCSI Commands Passed Through	No	No	Yes `REPORT LUNs` is not passed through
vCenter Server Support	Yes	Yes	Yes
Snapshots	Yes	Yes	No
Distributed Locking	Yes	Yes	Yes
Clustering	Cluster-in-a-box only	Cluster-in-a-box cluster-across-boxes	Physical-to-virtual clustering cluster-across-boxes
SCSI Target-Based Software	No	No	Yes

Use virtual disk files for the cluster-in-a-box type of clustering. If you plan to reconfigure your cluster-in-a-box clusters as cluster-across-boxes clusters, use virtual mode RDMs for the cluster-in-a-box clusters.

Create vSphere Virtual Machines with RDMs

When you give your vSphere virtual machine direct access to a raw SAN LUN, you create an RDM disk that resides on a VMFS datastore and points to the LUN. You can create the RDM as

an initial disk for a new virtual machine or add it to an existing virtual machine. When creating the RDM, you specify the LUN to be mapped and the datastore on which to put the RDM.

Although the RDM disk file has the same `.vmdk` extension as a regular virtual disk file, the RDM contains only mapping information. The actual virtual disk data is stored directly on the LUN.

This procedure assumes that you are creating a new virtual machine. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

1 Create a virtual machine.

   a Right-click any inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host, and select **New Virtual Machine**.

   b Select **Create a new virtual machine** and click **Next**.

   c Follow the steps required to create a virtual machine.

2 On the Customize Hardware page, click the **Virtual Hardware** tab.

3 (Optional) To delete the default virtual hard disk that the system created for your virtual machine, move your cursor over the disk and click the **Remove** icon.

4 Add an RDM disk.

   a Click **Add New Devices** and select **RDM Disk** from the list.

   b From the list of LUNs, select a target raw LUN and click **OK**.

   The system creates an RDM disk that maps your virtual machine to the target LUN. The RDM disk is shown on the list of virtual devices as a new hard disk.

5 Configure the RDM disk.

   a Click the **New Hard Disk** triangle to expand the properties for the RDM disk.

   b Select a location for the RDM.

   You can place the RDM on the same datastore where your virtual machine configuration files reside, or select a different datastore.

   Note To use vMotion for virtual machines with enabled NPIV, make sure that the RDM files and the virtual machine files are located on the same datastore. You cannot perform Storage vMotion when NPIV is enabled.

c Select a compatibility mode.

Option	Description
Physical	Allows the guest operating system to access the hardware directly. Physical compatibility is useful if you are using SAN-aware applications on the virtual machine. However, a virtual machine with a physical compatibility RDM cannot be cloned, made into a template, or migrated if the migration involves copying the disk.
Virtual	Allows the RDM to behave as if it were a virtual disk, so you can use such features as taking snapshots, cloning, and so on. When you clone the disk or make a template out of it, the contents of the LUN are copied into a `.vmdk` virtual disk file. When you migrate a virtual compatibility mode RDM, you can migrate the mapping file or copy the contents of the LUN into a virtual disk.

d If you selected virtual compatibility mode, select a disk mode.

Disk modes are not available for RDM disks using physical compatibility mode.

Option	Description
Dependent	Dependent disks are included in snapshots.
Independent - Persistent	Disks in persistent mode behave like conventional disks on your physical computer. All data written to a disk in persistent mode are written permanently to the disk.
Independent - Nonpersistent	Changes to disks in nonpersistent mode are discarded when you power off or reset the virtual machine. With nonpersistent mode, you can restart the virtual machine with a virtual disk in the same state every time. Changes to the disk are written to and read from a redo log file that is deleted when you power off or reset.

6 Complete your virtual machine configuration.

Deactivate SCSI INQUIRY Cache for Improved Performance in vSphere

Certain virtual machines with RDMs must obtain the SCSI INQUIRY information from the LUN instead of using SCSI INQUIRY data cached by ESXi.

Problem

Certain guest operating systems or applications that run in the virtual machines with the RDMs display unpredictable behavior.

Cause

This behavior might be caused by cached SCSI INQUIRY data that interferes with specific guest operating systems and applications.

When the ESXi host first connects to a target storage device, it issues the SCSI INQUIRY command to obtain basic identification data from the device. By default, ESXi caches the received SCSI INQUIRY data (Standard, page 80, and page 83), and the data remains unchanged afterwards. Responses for subsequent SCSI INQUIRY commands are returned from the cache.

However, specific guest operating systems running in virtual machines with RDMs must query the LUN instead of using SCSI INQUIRY data cached by ESXi. In these cases, you can configure the VM to ignore the SCSI INQUIRY cache.

Solution

- Use one of the following methods.

Option	Description
Modify the .vmx file of the virtual machine with the RDM	Use this method for the VMs with hardware version 8 or later. a Add the following parameter to the file: `scsix:y.ignoreDeviceInquiryCache = "true"` where *x* is the SCSI controller number and *y* is the SCSI target number of the RDM. b Reboot the VM.
Use the `esxcli` command	Because you configure the setting at a host level, no VM hardware version limitations apply. `esxcli storage core device inquirycache set --device device id --ignore true` No VM reboot is required.

No matter which method you use to set the SCSI INQUIRY cache parameter to true, the VM starts contacting the LUN directly for the SCSI INQUIRY data.

ignoreDeviceInquiryCache parameter in vmx	ignore inquirycache parameter in esxcli	Inquiry request serviced from
True	True	LUN
False (default if the parameter is not present)	True	LUN
True	False	LUN
False (default if the parameter is not present)	False	Cache

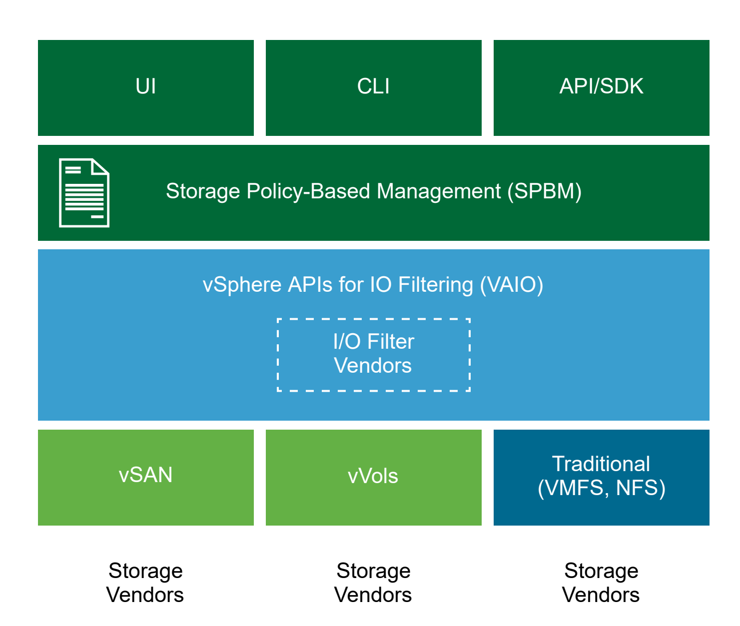
Storage Policy Based Management in vSphere

12

In vSphere, Storage Policy Based Management (SPBM) plays a major role by helping to align storage with application demands of your virtual machines. It provides a storage policy framework that serves as a single unified control panel across a broad range of data services and storage solutions.

As an abstraction layer, SPBM abstracts storage services delivered by Virtual Volumes, vSAN, I/O filters, or other storage entities.

Rather than integrating with each individual type of storage and data services, SPBM provides a universal framework for different types of storage entities.



SPBM offers the following mechanisms:

- Advertisement of storage capabilities and data services that storage arrays and other entities, such as I/O filters, offer.
- Bidirectional communications between ESXi and vCenter Server on one side, and storage arrays and entities on the other.
- Virtual machine provisioning based on VM storage policies.

About Virtual Machine Storage Policies

Virtual machine storage policies are essential to virtual machine provisioning through SPBM. The policies control which type of storage is provided for the virtual machine and how the virtual machine is placed within storage. They also determine data services that the virtual machine can use.

vSphere offers default storage policies. See Default Storage Policies in vSphere.

In addition, you can define policies and assign them to the virtual machines. See Workflow for vSphere VM Storage Policies.

You use the VM Storage Policies interface to create a storage policy. When you define the policy, you specify various storage requirements for applications that run on the virtual machines. You can also use storage policies to request specific data services, such as caching or replication, for virtual disks.

You apply the storage policy when you create, clone, or migrate the virtual machine. After you apply the storage policy, the SPBM mechanism assists you with placing the virtual machine in a matching datastore. In certain storage environments, SPBM determines how the virtual machine storage objects are provisioned and allocated within the storage resource to guarantee the required level of service. The SPBM also enables requested data services for the virtual machine and helps you to monitor policy compliance.

Read the following topics next:

- Default Storage Policies in vSphere
- Workflow for vSphere VM Storage Policies
- Populating the VM Storage Policies Interface in vSphere
- About Rules and Rule Sets in vSphere Storage Policies
- Creating and Managing vSphere VM Storage Policies
- About vSphere Storage Policy Components
- vSphere Storage Policies and Virtual Machines

Default Storage Policies in vSphere

When you provision a virtual machine on a datastore in vSphere, you must assign to the virtual machine a compatible VM storage policy. If you do not configure and explicitly assign the storage policy to the virtual machine, the system uses a default storage policy.

VMware-Provided Default Storage Policy

The generic default storage policy that ESXi provides applies to all datastores and does not include rules specific to any storage type.

In addition, ESXi offers the default storage policies for object-based datastores, vSAN or Virtual Volumes. These policies guarantee the optimum placement for the virtual machine objects within the object-based storage.

For information about the default storage policy for Virtual Volumes, see Virtual Volumes and VM Storage Policies.

VMFS and NFS datastores do not have specific default policies and can use the generic default policy or a custom policy you define for them.

User-Defined Default Storage Policies

You can create a VM storage policy that is compatible with vSAN or Virtual Volumes. You can then designate this policy as the default for vSAN and Virtual Volumes datastores. The user-defined default policy replaces the default storage policy that VMware provides.

Each vSAN and Virtual Volumes datastore can have only one default policy at a time. However, you can create a single storage policy with multiple placement rule sets, so that it matches multiple vSAN and Virtual Volumes datastores. You can designate this policy as the default policy for all datastores.

When the VM storage policy becomes the default policy for a datastore, you cannot delete the policy unless you disassociate it from the datastore.

Change the Default Storage Policy for a Datastore

For Virtual Volumes and vSAN datastores, VMware provides storage policies that are used as the default during the virtual machine provisioning. You can change the default storage policy for a selected Virtual Volumes or vSAN datastore.

Note Do not designate a storage policy with replication rules as a default storage policy. Otherwise, the policy prevents you from selecting replication groups.

Prerequisites

Create a storage policy that is compatible with Virtual Volumes or vSAN. You can create a policy that matches both types of storage.

Procedure

1. In the vSphere Client, navigate to the datastore.
2. Click the **Configure** tab, and click **General**.
3. In the Default Storage Policy pane, click **Edit**.
4. From the list of available storage policies, select a policy to designate as the default and click **OK**.

Results

The selected storage policy becomes the default policy for the datastore. The system assigns this policy to any virtual machine objects that you provision on the datastore when no other policy is selected.

Workflow for vSphere VM Storage Policies

The entire process of creating and managing storage policies in vSphere typically includes several steps. Whether you must perform a specific step might depend on the type of storage or data services that your environment offers.

Step	Description
Populate the VM Storage Policies interface with appropriate data.	The VM Storage Policies interface is populated with information about datastores and data services that are available in your storage environment. This information is obtained from storage providers and datastore tags. ■ For entities represented by storage providers, verify that an appropriate provider is registered. Entities that use the storage provider include vSAN, Virtual Volumes, and I/O filters. Depending on the type of storage entity, some providers are self-registered. Other providers must be manually registered. See Use Storage Providers to Populate the VM Storage Policies Interface and Register Storage Providers for Virtual Volumes. ■ Tag datastores that are not represented by storage providers. You can also use tags to indicate a property that is not communicated through the storage provider, such as geographical location or administrative group. See Assign Tags to Datastores.
Create predefined storage policy components.	A storage policy component describes a single data service, such as replication, that must be provided for the virtual machine. You can define the component in advance and associate it with multiple VM storage policies. The components are reusable and interchangeable. See Create Storage Policy Components.
Create VM storage policies.	When you define storage policies for virtual machines, you specify storage requirements for applications that run on the virtual machines. See Creating and Managing vSphere VM Storage Policies.
Apply the VM storage policy to the virtual machine.	You can apply the storage policy when deploying the virtual machine or configuring its virtual disks. See Assign Storage Policies to Virtual Machines.
Check compliance for the VM storage policy.	Verify that the virtual machine uses the datastore that is compliant with the assigned storage policy. See Check Compliance for a VM Storage Policy.

To create and manage your storage policies, you use the VM Storage Policy interface of the vSphere Client.

Populating the VM Storage Policies Interface in vSphere

Before you start creating VM storage policies, you must populate the VM Storage Policy interface with information about storage entities and data services that are available in your vSphere storage environment.

This information is obtained from storage providers, also called VASA providers. Another source is datastore tags.

Storage Capabilities and Services

Certain datastores, for example, Virtual Volumes and vSAN, are represented by the storage providers. Through the storage providers, the datastores can advertise their capabilities in the VM Storage Policy interface. These datastore capabilities, data services, and other characteristics with ranges of values populate the VM Storage Policy interface.

You use these characteristics when you define datastore-based placement and service rules for your storage policy.

Data Services

I/O filters on your hosts are also represented by the storage providers. The storage provider delivers information about the data services of the filters to the VM Storage Policy interface. You use this information when defining the rules for host-based data services, also called common rules. Unlike the datastore-specific rules, these rules do not define storage placement and storage requirements for the virtual machine. Instead, they activate the requested I/O filter data services for the virtual machine.

Tags

Generally, VMFS and NFS datastores are not represented by a storage provider. They do not display their capabilities and data services in the VM Storage Polices interface. You can use tags to encode information about these datastores. For example, you can tag your VMFS datastores as VMFS-Gold and VMFS-Silver to represent different levels of service.

For Virtual Volumes and vSAN datastores, you can use tags to encode information that is not advertised by the storage provider, such as geographical location (Palo Alto), or administrative group (Accounting).

Similar to the storage capabilities and characteristics, all tags associated with the datastores appear in the VM Storage Policies interface. You can use the tags when you define the tag-based placement rules.

Use Storage Providers to Populate the VM Storage Policies Interface

For entities represented by storage (VASA) providers, verify that an appropriate provider is registered. After the storage providers are registered, the VM Storage Policies interface becomes populated with information about datastores and data services that the providers represent.

Entities that use the storage provider include vSAN, Virtual Volumes, and I/O filters. Depending on the type of the entity, some providers are self-registered. Other providers, for example, the Virtual Volumes storage provider, must be manually registered. After the storage providers are registered, they deliver the following data to the VM Storage Policies interface:

- Storage capabilities and characteristics for such datastores as Virtual Volumes and vSAN.
- Data services the I/O filters provide.

Prerequisites

Register the storage providers that require manual registration. For more information, see the appropriate documentation:

- *Administering VMware vSAN*
- Chapter 14 Working with VMware vSphere Virtual Volumes
- Chapter 15 Filtering Virtual Machine I/O in vSphere

Procedure

1 Browse to the vCenter Server instance.

2 Click the **Configure** tab, and click **Storage Providers**.

3 In the Storage Providers list, view the storage providers registered with vCenter Server.

   The list shows general information including the name of the storage provider, its URL and status, storage entities that the provider represents, and so on.

4 To display more details, select a specific storage provider or its component from the list.

Assign Tags to Datastores

Use tags to encode information about a datastore. The tags are helpful when your datastore is not represented by a storage provider and does not advertise its services in the VM Storage Polices interface. You can also use the tags to indicate a property that is not communicated through a storage provider, such as a geographical location or administrative group.

You can apply a new tag that contains general storage information to a datastore. For more details about the tags, their categories, and how to manage the tags, see the *vCenter Server and Host Management* documentation.

Prerequisites

Required privileges:

- **vSphere Tagging.Create vSphere Tag** on the root vCenter Server instance
- **vSphere Tagging.Create vSphere Tag Category** on the root vCenter Server instance
- **vSphere Tagging.Assign or Unassign vSphere Tag** on the root vCenter Server instance

Procedure

1 In the vSphere Client, create a category for storage tags.

   a From the Home menu, click **Tags & Custom Attributes**.

   b Click the **Tags** tab and click **Categories**.

   c Click the **Add Category** icon.

   d Specify the category properties. See the following example.

Category Property	Example
Category Name	Storage Location
Description	Category for tags related to location of storage
Tags Per Object	**Many tags**
Associable Object Types	**Datastore** and **Datastore Cluster**

   e Click **OK**.

2 Create a storage tag.

   a On the **Tags** tab, click **Tags**.

   b Click the **Add Tag** icon.

   c Specify the properties for the tag. See the following example.

Tag Property	Example
Name	Texas
Description	Datastore located in Texas
Category	Storage Location

   d Click **OK**.

3 Apply the tag to the datastore.

   a Navigate to the datastore.

   b Right-click the datastore, and select **Tags & Custom Attributes > Assign Tag**.

   c From the list of tags, select an appropriate tag, for example, Texas in the Storage Location category, and click **Assign**.

Results

The new tag is assigned to the datastore and appears on the datastore **Summary** tab in the **Tags** pane.

What to do next

When creating a VM storage policy, you can reference the tag to include the tagged datastore in the list of compatible storage resources. See Create a VM Storage Policy for Tag-Based Placement.

Or you can exclude the tagged datastore from the VM storage policy. For example, your VM storage policy can include Virtual Volumes datastores located in Texas and California, but exclude datastores located in Nevada.

To learn more about how to use tags in VM storage policies, watch the following video.



(Using Tags in Storage Policies)

About Rules and Rule Sets in vSphere Storage Policies

After the VM Storage Policies interface is populated with the appropriate data, you can start creating your vSphere storage policies. Creating a policy involves defining specific storage placement rules and rules to configure data services.

Rules

The rule is a basic element of the VM storage policy. Each individual rule is a statement that describes a single requirement for virtual machine storage and data services.

Rule Sets

Within a storage policy, individual rules are organized into collections of rules, or rule sets. Typically, the rule sets can be in one of the following categories: rules for host-based services and datastore-specific rules.

Datastore-Specific Rule Sets

Each rule set must include placement rules that describe requirements for virtual machine storage resources. All placement rules within a single rule set represent a single storage entity. These rules can be based on storage capabilities or tags.

In addition, the datastore-specific rule set can include optional rules or storage policy components that describe data services to provide for the virtual machine. Generally, these rules request such services as caching, replication, other services provided by storage systems.

To define the storage policy, one datastore-specific set is required. Additional rule sets are optional. A single policy can use multiple sets of rules to define alternative storage placement parameters, often from several storage providers.

Placement Rules: Capability-Based

Placement rules specify a particular storage requirement for the VM and enable SPBM to distinguish compatible datastores among all datastores in the inventory. These rules also

describe how the virtual machine storage objects are allocated within the datastore to receive the required level of service. For example, the rules can list Virtual Volumes as a destination and define the maximum recovery point objective (RPO) for the Virtual Volumes objects.

When you provision the virtual machine, these rules guide the decision that SPBM makes about the virtual machine placement. SPBM finds the Virtual Volumes datastores that can match the rules and satisfy the storage requirements of the virtual machine. See Create a VM Storage Policy for Virtual Volumes.

Placement Rules: Tag-Based

Tag-based rules reference datastore tags. These rules can define the VM placement, for example, request as a target all datastores with the VMFS-Gold tag. You can also use the tag-based rules to fine-tune your VM placement request further. For example, exclude datastores with the Palo Alto tag from the list of your Virtual Volumes datastores. See Create a VM Storage Policy for Tag-Based Placement.

Rules for Host-Based Services

This rule set activates data services provided by the host. The set for host-based services can include rules or storage policy components that describe particular data services, such as encryption or replication.

Unlike datastore-specific rules, this set does not include placement rules. Rules for host-based services are generic for all types of storage and do not depend on the datastore. See Create a VM Storage Policy for Host-Based Data Services.

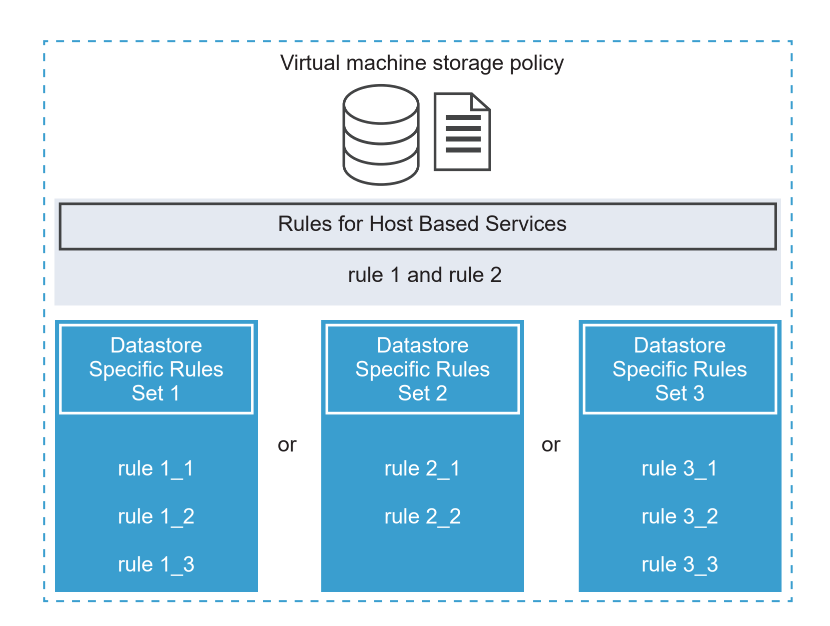
Table 12-1. Structure of a VM Storage Policy

Rules for Host-Based Services	Datastore-Specific Rule Sets
Rules or predefined storage policy components to activate data services installed on ESXi hosts. For example, replication by I/O filters.	Capability-based or tag-based placement rules that describe requirements for virtual machine storage resources. For example, Virtual Volumes placement.
	Rules or predefined storage policy components that activate data services provided by storage. For example, caching by Virtual Volumes.

Relationships Between Rules and Rule Sets

The boolean operator `OR` defines the relationship between the datastore-specific rule sets within the policy. The `AND` operator defines the relationship between all rules within a single rule set. The policy can contain only a rule set for host-based services, or only a datastore-specific rule set, or both.

If the rule set for host-based services is not present, meeting all the rules of a single datastore-specific rule set is sufficient to satisfy the entire policy. If the rule set for host-based services is present, the policy matches the datastore that satisfies the host services rules and all rules in one of the datastore-specific sets.



Creating and Managing vSphere VM Storage Policies

To create and manage storage policies for your virtual machines in vSphere, you use the VM Storage Policies interface.

Create a VM Storage Policy for VMFS Datastore

Create a custom storage policy compatible with VMFS datastores. In the storage policy, you can specify disk or volume provisioning options.

Prerequisites

- Be familiar with provisioning policies. See About Virtual Disk Provisioning Policies.
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

1 Open the **Create VM Storage Policy** wizard.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**.

   c Click **Create**.

2 Enter the policy name and description, and click **Next**.

Option	Action
vCenter Server	Select the vCenter Server instance.
Name	Enter the name of the storage policy, for example VMFS Storage Policy.
Description	Enter the description of the storage policy.

3 On the **Policy structure** page under **Datastore specific rules**, enable rules for a target storage entity, VMFS storage.

   You can enable rules for several datastores. Multiple rule sets allow a single policy to define alternative storage placement parameters, often from several storage providers.

4 On the **VMFS** rules page, define storage placement rules for the target VMFS datastore.

   a Click the **Placement** tab.

   b From the **Volume allocation** drop-down menu, select one of the following options.

Option	Description
Conserve space when possible	Use to create a virtual disk of volume in thin format. The option neither guarantees space reservation nor runtime performance. This option gives fast provisioning time but the space is allocated on demand and the blocks are also initialized on demand.
Fully initialized	Use to create a disk or volume in thick provision eager zeroed format. The option guarantees that the required space is reserved on the datastore and all the blocks of the disk are zeroed. This option gives the best runtime performance, but provisioning time is proportional to the size of the disk.
Reserve space	Use to create a disk or volume in thick provision lazy zeroed format. The option guarantees that the required space is reserved on the datastore. The blocks of the disk are not zeroed, so there are no guarantee on the runtime performance.

   c To fine-tune your placement request further, click the **Tags** tab and add a tag-based rule.

   Tag-based rules can filter datastores by including or excluding specific placement criteria. For example, your VM storage policy can include VMFS datastores located in Texas and California, but exclude datastores located in Nevada.

Create a VM Storage Policy for Host-Based Data Services

To define the VM storage policy in the vSphere Client, use the **Create VM Storage Policy** wizard. In this task, you create rules for data services offered by ESXi hosts. The VM storage policy that includes these rules activates specified data services for the virtual machine.

Available data services include encryption, I/O control, caching, and so on. Certain data services, such as encryption, are provided by VMware. Others can be offered by third-party I/O filters that you install on your host.

The data services are usually generic for all types of storage and do not depend on a datastore. Adding datastore-specific rules to the storage policy is optional.

If you add datastore-specific rules, and both the I/O filters on the host and storage offer the same type of service, for example, encryption, your policy can request this service from both providers. As a result, the virtual machine data is encrypted twice, by the I/O filter and your storage. However, replication provided by Virtual Volumes and replication provided by the I/O filter cannot coexist in the same storage policy.

Prerequisites

- For information about encrypting your virtual machines, see the *vSphere Security* documentation.
- For information about I/O filters, see Chapter 15 Filtering Virtual Machine I/O in vSphere.
- For information about storage policy components, see About vSphere Storage Policy Components.
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

1 Open the **Create VM Storage Policy** wizard.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**

   c Click **Create**.

2 Enter the policy name and description, and click **Next**.

Option	Action
vCenter Server	Select the vCenter Server instance.
Name	Enter the name of the storage policy.
Description	Enter the description of the storage policy.

3 On the **Policy structure** page under **Host based services**, click **Enable host based rules**.

4 On the **Host based services** page, define rules and configure data services provided by your host.

   a Click the tab for the data service category, for example, **Encryption**.

   b Define custom rules for the data service category or use predefined components.

Option	Description
Disabled	This is a default setting.
Use storage policy component	Select a storage policy component from the drop-down menu. This option is available only if you have predefined components in your database.
Custom	Define custom rules for the data service category by specifying an appropriate provider and values for the rules.

Note You can activate several data services. If you use encryption with other data services, use the custom encryption option and set the **Allow I/O filters before encryption** parameter to **True**. The parameter ensures that other services, such as replication, can analyze clear text data before it is encrypted.

5 On the **Storage compatibility** page, review the list of datastores that match this policy.

To be compatible with the policy for host-based services, datastores must be connected to the host that provides these services. If you add datastore-specific rule sets to the policy, the compatible datastores must also satisfy storage requirements of the policy.

6 On the **Review and finish** page, review the storage policy settings and click **Finish**.

To change any settings, click **Back** to go to the relevant page.

Results

The new VM storage policy for host-based data services appears on the list.

Create a VM Storage Policy for Virtual Volumes

To define the VM storage policy in the vSphere Client, use the **Create VM Storage Policy** wizard. In this task, you create a custom storage policy compatible with Virtual Volumes. When you define the VM storage policy for Virtual Volumes, you create rules to configure storage and data services provided by the Virtual Volumes datastore. The rules are applied when the VM is placed on the Virtual Volumes datastore. The custom storage policy can replace the default No Requirements storage policy for Virtual Volumes that VMware provides.

The procedure assumes that you are creating the VM storage policy for Virtual Volumes. For information about the vSAN storage policy, see the *Administering VMware vSAN* documentation.

Prerequisites

- Verify that the Virtual Volumes storage provider is available and active. See Register Storage Providers for Virtual Volumes.
- Make sure that the VM Storage Policies interface is populated with information about storage entities and data services that are available in your storage environment. See Populating the VM Storage Policies Interface in vSphere.
- Define appropriate storage policy components. See Create Storage Policy Components.
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

1 Open the **Create VM Storage Policy** wizard.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**.

   c Click **Create**.

2 Enter the policy name and description, and click **Next**.

Option	Action
vCenter Server	Select the vCenter Server instance.
Name	Enter the name of the storage policy, for example Virtual Volumes Storage Policy.
Description	Enter the description of the storage policy.

3 On the **Policy structure** page under Datastore specific rules, enable rules for a target storage entity, such as Virtual Volumes storage.

You can enable rules for several datastores. Multiple rule sets allow a single policy to define alternative storage placement parameters, often from several storage providers.

4 On the *Virtual Volumes* rules page, define storage placement rules for the target Virtual Volumes datastore.

a Click the **Placement** tab and click **Add Rule**.

b From the Add Rule drop-down menu, select available capability and specify its value.

For example, you can specify the number of read operations per second for the Virtual Volumes objects.

You can include as many rules as you need for the selected storage entity. Verify that the values you provide are within the range of values that the Virtual Volumes datastore advertises.

c To fine-tune your placement request further, click the **Tags** tab and add a tag-based rule.

Tag-based rules can filter datastores by including or excluding specific placement criteria. For example, your VM storage policy can include Virtual Volumes datastores located in Texas and California, but exclude datastores located in Nevada.

5 (Optional) Define rules to configure datastore-specific services.

The data services, such as encryption, caching, or replication, are offered by the storage. The VM storage policy that references data services, requests these services for the VM when the VM is placed to the Virtual Volumes datastore.

a Click the tab for the data service category, for example, **Replication**.

b Define custom rules for the data service category or use predefined components.

Option	Description
Disabled	This is a default setting.
Use storage policy component	Select a storage policy component from the drop-down menu. This option is available only if you have predefined components in your database.
Custom	Define custom rules for the data service category by specifying an appropriate provider and values for the rules.

6 On the **Storage compatibility** page, review the list of datastores that match this policy.

   If the policy includes several rule sets, the datastore must satisfy at least one rule set and all rules within this set.

7 On the **Review and finish** page, review the storage policy settings and click **Finish**.

   To change any settings, click **Back** to go to the relevant page.

Results

The new VM storage policy compatible with Virtual Volumes appears on the list.

What to do next

You can now associate this policy with a virtual machine, or designate the policy as default.

Create a VM Storage Policy for Tag-Based Placement

Tag-based rules reference the tags that you assign to the datastores and can filter the datastores to be used for placement of the VMs. To define tag-based placement in the vSphere Client, use the **Create VM Storage Policy** wizard.

Note At a datastore cluster level, you can use a tag-based storage policy only when all datastores in the cluster have consistent tags.

Prerequisites

- Make sure that the VM Storage Policies interface is populated with information about storage entities and data services that are available in your storage environment. See Populating the VM Storage Policies Interface in vSphere.
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

1 Open the **Create VM Storage Policy** wizard.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**.

   c Click **Create**.

2 Enter the policy name and description, and click **Next**.

Option	Action
vCenter Server	Select the vCenter Server instance.
Name	Enter the name of the storage policy.
Description	Enter the description of the storage policy.

3 On the **Policy structure** page under Datastore specific rules, enable tag-based placement rules.

4 On the **Tag based placement** page, create the tag rules.

   a Click **Add Tag Rule** and define tag-based placement criteria. Use the following as an example.

Option	Example
Tag category	Level of Service
Usage option	Use storage tagged with
Tags	Gold

   All datastores with the Gold tag become compatible as the storage placement target.

   b (Optional) Add more tag-based rules.

5 On the **Storage compatibility** page, review the list of datastores that match this policy.

6 On the **Review and finish** page, review the storage policy settings and click **Finish**.

   To change any settings, click **Back** to go to the relevant page.

Results

The new VM storage policy compatible with tagged datastores appears on the list.

Edit or Clone a VM Storage Policy

If storage requirements for virtual machines and virtual disks change, you can modify the existing storage policy. You can also create a copy of the existing VM storage policy by cloning it. While cloning, you can optionally select to customize the original storage policy.

Prerequisites

Required privilege: **StorageProfile.View**

Procedure

1 In the vSphere Client, navigate to the storage policy.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**.

2 Select the storage policy, and click one of the following icons:

   - **Edit**
   - **Clone**

3 (Optional) Modify the policy and click **OK**.

4 If editing the storage policy that is used by a virtual machine, reapply the policy to the virtual machine.

Option	Description
Manually later	If you select this option, the compliance status for all virtual disks and virtual machine home objects associated with the storage policy changes to Out of Date. To update configuration and compliance, manually reapply the storage policy to all associated entities. See Reapply Virtual Machine Storage Policy.
Now	Update virtual machine and compliance status immediately after editing the storage policy.

About vSphere Storage Policy Components

A VM storage policy in vSphere can include one or several reusable and interchangeable building blocks, called storage policy components. Each component describes a particular data service to be provided for the virtual machine. You can define the policy components in advance and associate them with multiple VM storage policies.

You cannot assign the predefined component directly to a virtual machine or virtual disk. Instead, you must add the component to the VM storage policy, and assign the policy to the virtual machine.

The component describes one type of service from one service provider. The services can vary depending on the providers that you use, but generally belong in one of the following categories.

- Compression
- Caching
- Encryption
- Replication

When you create the storage policy component, you define the rules for one specific type and grade of service.

The following example shows that virtual machines VM1 and VM2 have identical placement requirements, but must have different grades of replication services. You can create the storage policy components with different replication parameters and add these components to the related storage policies.

Table 12-2. Storage Policy Components

Virtual Machine	Placement Rules	Storage Policy Component
VM1 requires replication every 2 hours	Virtual Volumes Datastore	2-hour Replication
VM2 requires replication every 4 hours	Virtual Volumes Datastore	4-hour Replication

The provider of the service can be a storage system, an I/O filter, or another entity. If the component references an I/O filter, the component is added to the host-based rules of the storage policy. Components that reference entities other than the I/O filters, for example, a storage system, are added to the datastore-specific rule sets.

When you work with the components, follow these guidelines:

- Each component can include only one set of rules. All characteristics in this rule set belong to a single provider of the data services.
- If the component is referenced in the VM storage policy, you cannot delete the component. Before deleting the component, you must remove it from the storage policy or delete the storage policy.
- When you add components to the policy, you can use only one component from the same category, for example caching, per a set of rules.

Create Storage Policy Components

A storage policy component describes a single data service, such as replication, that must be provided for the virtual machine. You can define the component in advance and associate it with multiple VM storage policies. The components are reusable and interchangeable.

Procedure

1 In the vSphere Client, open the **New Storage Policy Component** dialog box.

    a Click **Menu > Policies and Profiles**.

    b Under **Policies and Profiles**, click **Storage Policy Components**.

2 Click **Create Storage Policy Component**.

3 Select the vCenter Server instance.

4 Enter a name, for example, 4-hour Replication, and a description for the policy component.

    Make sure that the name does not conflict with the names of other components or storage policies.

5 Select the category of the service, for example, **Replication**.

6 Select the service provider.

7 Define rules for the selected category.

    For example, if you are configuring 4-hour replication, set the Recovery Point Objective (RPO) value to 4.

For encryption based on I/O filters, set the **Allow I/O filters before encryption** parameter. Encryption provided by storage does not require this parameter.

Option	Description
False (default)	Does not allow the use of other I/O filters before the encryption filter.
True	Allows the use of other I/O filters before the encryption filter. Other filters, such as replication, can analyze clear text data before it is encrypted.

8 Click **OK**.

Results

The new component appears in the list of storage policy components.

What to do next

You can add the component to the VM storage policy. If the data service that the component references is provided by the I/O filters, you add the component to the host-based rules of the storage policy. Components that reference entities other than the I/O filters, for example, a storage system, are added to the datastore-specific rule sets.

Edit or Clone Storage Policy Components

You can modify the existing storage policy components. You can also create a copy of the existing component by cloning it.

Procedure

1 In the vSphere Client, navigate to the storage policy component to edit or clone.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **Storage Policy Components**.

2 Select the component and click one of the following icons.

Option	Description
Edit Settings	When editing, you cannot change the category of the data service and the provider. For example, if the original component references replication provided by I/O filters, these settings must remain unchanged.
Clone	When cloning, you can customize any settings of the original component.

3 Modify appropriate values, and click **OK**.

4 If a VM storage policy that is assigned to a virtual machine references the policy component you edit, reapply the storage policy to the virtual machine.

Menu Item	Description
Manually later	If you select this option, the compliance status for all virtual disks and virtual machine home objects associated with the storage policy changes to Out of Date. To update configuration and compliance, manually reapply the storage policy to all associated entities. See Reapply Virtual Machine Storage Policy.
Now	Update virtual machine and compliance status immediately after editing the storage policy.

vSphere Storage Policies and Virtual Machines

After you define a VM storage policy in vSphere, you can apply it to a virtual machine. You apply the storage policy when provisioning the virtual machine or configuring its virtual disks. Depending on its type and configuration, the policy might serve different purposes. The policy can select the most appropriate datastore for the virtual machine and enforce the required level of service. Or it can enable specific data services for the virtual machine and its disks.

If you do not specify the storage policy, the system uses a default storage policy that is associated with the datastore. If your storage requirements for the applications on the virtual machine change, you can modify the storage policy that was originally applied to the virtual machine.

Assign Storage Policies to Virtual Machines

You can assign a VM storage policy in an initial deployment of a virtual machine or when you perform other virtual machine operations, such as cloning or migrating.

This topic describes how to assign the VM storage policy when you create a virtual machine. For information about other deployment methods that include cloning, deployment from a template, and so on, see the *vSphere Virtual Machine Administration* documentation.

You can apply the same storage policy to the virtual machine configuration file and all its virtual disks. If storage requirements for your virtual disks and the configuration file are different, you can associate different storage policies with the VM configuration file and the selected virtual disks.

Procedure

1 Start the virtual machine provisioning process and follow the appropriate steps.

2 Assign the same storage policy to all virtual machine files and disks.

   a On the **Select storage** page, select a storage policy from the **VM Storage Policy** drop-down menu.

      Based on its configuration, the storage policy separates all datastores into compatible and incompatible. If the policy references data services offered by a specific storage entity, for example, Virtual Volumes, the compatible list includes datastores that represent only that type of storage.

   b Select an appropriate datastore from the list of compatible datastores.

      The datastore becomes the destination storage resource for the virtual machine configuration file and all virtual disks.

   c If you use the replication service with Virtual Volumes, specify the replication group.

      Replication groups indicate which VMs and virtual disks must be replicated together to a target site.

Option	Description
Preconfigured replication group	Replication groups that are configured in advance on the storage side. vCenter Server and ESXi discover the replication groups, but do not manage their life cycle.
Automatic replication group	Virtual Volumes creates a replication group and assigns all VM objects to this group.

3 Change the VM storage policy for the virtual disk.

   Use this option if requirements for storage placement are different for virtual disks. You can also use this option to enable I/O filter services, such as caching and replication, for your virtual disks.

   a On the **Customize hardware** page, expand the **New hard disk** pane.

   b From the **VM storage policy** drop-down menu, select the storage policy to assign to the virtual disk.

   c (Optional) Change the storage location of the virtual disk.

      Use this option to store the virtual disk on a datastore other than the datastore where the VM configuration file resides.

4 Complete the virtual machine provisioning process.

Results

After you create the virtual machine, the **Summary** tab displays the assigned storage policies and their compliance status.

What to do next

If storage placement requirements for the configuration file or the virtual disks change, you can later modify the virtual policy assignment.

Change Storage Policy Assignment for Virtual Machine Files and Disks

If your storage requirements for the applications on the virtual machine change, you can edit the storage policy that was originally applied to the virtual machine.

You can edit the storage policy for a powered-off or powered-on virtual machine.

When changing the VM storage policy assignment, you can apply the same storage policy to the virtual machine configuration file and all its virtual disks. You can also associate different storage policies with the VM configuration file and the virtual disks. You might apply different policies when, for example, storage requirements for your virtual disks and the configuration file are different.

Procedure

1 In the vSphere Client, browse to the virtual machine.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**.

   c Click the storage policy you want to change, and click **VM Compliance**.

   You can see the list of virtual machines that use this storage policy.

   d Click the virtual machine whose policy you want to modify.

2 Click the **Configure** tab and click **Policies**.

3 Click **Edit VM Storage Policies**.

4 Specify the VM storage policy for your virtual machine.

Option	Actions
Apply the same storage policy to all virtual machine objects	Select the policy from the **VM storage policy** drop-down menu.
Apply different storage policies to the VM home object and virtual disks	a Turn on the **Configure per disk** option. b Select the object, for example, VM home. c In the VM Storage Policy column, select the policy from the drop-down menu.

5 If you use Virtual Volumes policy with replication, configure the replication group.

   Replication groups indicate which VMs and virtual disks must be replicated together to a target site.

   All storage objects of a VM should belong to the same replication group. You cannot assign different replication groups to different storage objects of a VM.

6 Click **OK** to save the VM storage policy changes.

Results

The storage policy is assigned to the virtual machine and its disks.

Check Compliance for a VM Storage Policy

You can check whether a virtual machine uses a datastore that is compatible with the storage requirements specified in the VM storage policy.

Prerequisites

Verify that the virtual machine has a storage policy that is associated with it.

Procedure

1 In the vSphere Client, navigate to the virtual machine.

2 Click the **Configure** tab and click **Policies**.

3 Click **Check VM Storage Policy Compliance**.

   The system verifies the compliance.

4 View the compliance status.

Compliance Status	Description
Compliant	The datastore that the virtual machine or virtual disk uses has the storage capabilities compatible with the policy requirements.
Noncompliant	The datastore that the virtual machine or virtual disk uses does not have the storage capabilities compatible with the policy requirements. You can migrate the virtual machine files and virtual disks to compliant datastores.
Out of Date	The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore where the virtual machine objects reside. To communicate the changes, reapply the policy to the objects that are out of date.
Not Applicable	This storage policy references datastore capabilities that are not supported by the datastore where virtual machine resides.

What to do next

When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See Find Compatible Storage Resource for Noncompliant Virtual Machine.

If the status is Out of Date, reapply the policy to the objects. See Reapply Virtual Machine Storage Policy.

Find Compatible Storage Resource for Noncompliant Virtual Machine

Determine which datastore is compatible with the storage policy that is associated with your virtual machine.

Occasionally, a storage policy that is assigned to a virtual machine can be in the noncompliant status. This status indicates that the virtual machine or its disks use datastores that are incompatible with the policy. You can migrate the virtual machine files and virtual disks to compatible datastores.

Use this task to determine which datastores satisfy the requirements of the policy.

Procedure

1. Verify that the storage policy for the virtual machine is in the noncompliant state.
   a. In the vSphere Client, navigate to the virtual machine.
   b. Click the **Summary** tab.

      The VM Storage Policy Compliance panel on the VM Storage Policies pane shows the Noncompliant status.
2. Navigate to the noncompliant storage policy.
   a. Click **Menu > Policies and Profiles**.
   b. Under **Policies and Profiles**, click **VM Storage Policies**.
3. Display the list of compatible datastores for the noncompliant storage policy.
   a. Click the storage policy.
   b. Click **Storage Compatibility**.

   The list of datastores that match the requirements of the policy appears.

What to do next

You can migrate the virtual machine or its disks to one of the datastores in the list.

Reapply Virtual Machine Storage Policy

After you edit a storage policy that is already associated with a virtual machine object, you must reapply the policy. By reapplying the policy, you communicate new storage requirements to the datastore where the virtual machine object resides.

Prerequisites

The compliance status for a virtual machine is Out of Date. The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore.

Procedure

1. In the vSphere Client, navigate to the virtual machine.
2. Click the **Configure** tab and click **Policies**.
3. Verify that the compliance status is Out of Date.
4. Click **Reapply VM Storage Policy**.

5 Check the compliance status.

Compliance Status	Description
Compliant	The datastore that the virtual machine or virtual disk uses has the storage capabilities that the policy requires.
Noncompliant	The datastore supports specified storage requirements, but cannot currently satisfy the storage policy. For example, the status might become Noncompliant when physical resources of the datastore are unavailable. You can bring the datastore into compliance by making changes in the physical configuration of your host cluster. For example, by adding hosts or disks to the cluster. If additional resources satisfy the storage policy, the status changes to Compliant. When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See Find Compatible Storage Resource for Noncompliant Virtual Machine.
Not Applicable	The storage policy references datastore capabilities that are not supported by the datastore.

Using Storage Providers in vSphere

13

A storage provider is a software component that is offered by VMware or developed by a third party through vSphere APIs for Storage Awareness (VASA). The storage provider can also be called VASA provider. The storage providers integrate with various storage entities that include external physical storage and storage abstractions, such as vSAN and Virtual Volumes. Storage providers can also support software solutions, for example, I/O filters.

Storage Provider Concepts

Generally, vCenter Server and ESXi use the storage providers to obtain information about storage configuration, status, and storage data services offered in your environment. This information appears in the vSphere Client. The information helps you to make appropriate decisions about virtual machine placement, to set storage requirements, and to monitor your storage environment.

Persistence Storage Providers

Storage providers that manage arrays and storage abstractions, are called persistence storage providers. Providers that support Virtual Volumes or vSAN belong to this category. In addition to storage, persistence providers can provide other data services, such as replication.

Data Service Providers

Another category of providers is I/O filter storage providers, or data service providers. These providers offer data services that include host-based caching, compression, and encryption.

Both persistence storage and data service providers can belong to one of these categories.

Built-in Storage Providers

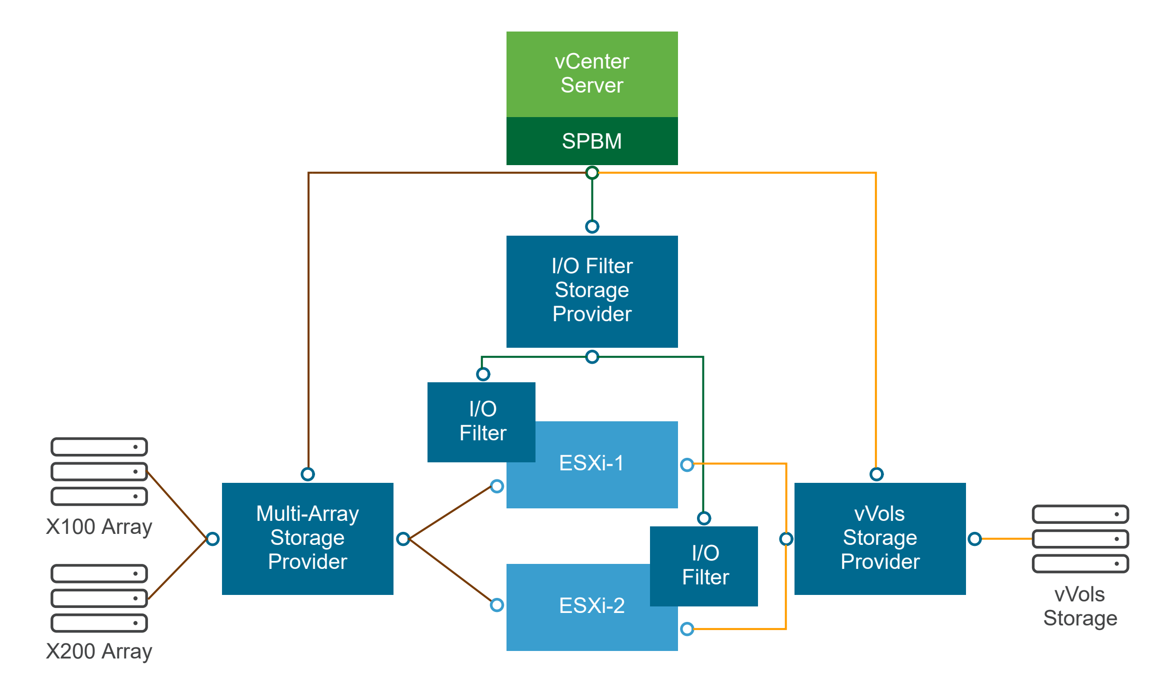
Built-in storage providers are offered by VMware. Typically, they do not require registration. For example, the storage providers that support vSAN or I/O filters are build-in and become registered automatically.

Third-Party Storage Providers

When a third party offers a storage provider, you typically must register the provider. An example of such a provider is the Virtual Volumes provider. You use the vSphere Client to register and manage each storage provider component.

For more information, Manage Storage Providers for vSphere Virtual Volumes.

The following graphic illustrates how different types of storage providers facilitate communications between vCenter Server and ESXi and other components of your storage environment. For example, the components might include storage arrays, Virtual Volumes storage, and I/O filters.



Storage Providers and Data Representation

Information that the storage provider supplies can be divided into the following categories:

- Storage data services and capabilities. This type of information is essential for such functionalities as vSAN, Virtual Volumes, and I/O filters. The storage provider that represents these functionalities integrates with the Storage Policy Based Management (SPBM) mechanism. The storage provider collects information about data services that are offered by underlying storage entities or available I/O filters.

  You reference these data services when you define storage requirements for virtual machines and virtual disks in a storage policy. Depending on your environment, the SPBM mechanism ensures appropriate storage placement for a virtual machine or enables specific data services for virtual disks. For details, see Creating and Managing vSphere VM Storage Policies.

- Storage status. This category includes reporting about status of various storage entities. It also includes alarms and events for notifying about configuration changes.

  This type of information can help you troubleshoot storage connectivity and performance problems. It can also help you to correlate array-generated events and alarms to corresponding performance and load changes on the array.

- Storage DRS information for the distributed resource scheduling on block devices or file systems. This information helps to ensure that decisions made by Storage DRS are compatible with resource management decisions internal to the storage systems.

Storage Provider Requirements and Considerations

When you use third-party storage provider, vendors are responsible for supplying storage providers. The VMware VASA program defines an architecture that integrates third-party storage providers into the vSphere environment, so that vCenter Server and ESXi hosts can communicate with the storage providers.

To use storage providers, follow these requirements:

- Make sure that every storage provider you use is certified by VMware and properly deployed. For information about deploying the storage providers, contact your storage vendor.
- Make sure that the storage provider is compatible with the vCenter Server and ESXi versions. See *VMware Compatibility Guide*.
- Do not install the VASA provider on the same system as vCenter Server.
- If your environment contains older versions of storage providers, existing functionality continues to work. However, to use new features, upgrade your storage provider to a new version.

Read the following topics next:

- Managing Storage Providers

Managing Storage Providers

You might need to register certain types of storage providers to establish their connection to vCenter Server. You can then review and manage the storage providers.

Register Storage Providers

Use the vSphere Client to register a separate storage provider for each host in a cluster.

After registration, vCenter Server can detect and use the functionality of the latest VASA version.

Note Built-in storage providers do not require registration. For example, the storage providers that support vSAN or I/O filters are build-in. They become registered and appear on the list of storage providers automatically. For more information, see the *Administering VMware vSAN* documentation and Chapter 15 Filtering Virtual Machine I/O in vSphere.

Prerequisites

Verify that the storage provider component is installed on the storage side and obtain its credentials from your storage administrator.

Procedure

1 Navigate to vCenter Server.

2 Click the **Configure** tab, and click **Storage Providers**.

3 Click the **Add** icon.

4 Enter connection information for the storage provider, including the name, URL, and credentials.

5 Click **OK**.

Results

vCenter Server registers the storage provider and establishes a secure connection with it.

What to do next

To troubleshoot registration of your storage provider, see the VMware Knowledge Base article https://kb.vmware.com/s/article/49798.

View Storage Provider Information

After you register a storage provider component with vCenter Server, the storage provider appears on the storage providers list. Certain storage providers are self-registered and automatically appear on the list after you set up the entity they represent, for example, vSAN or I/O filters.

Use the vSphere Client to view general storage provider information and details for each storage component.

Procedure

1 Navigate to vCenter Server.

2 Click the **Configure** tab, and click **Storage Providers**.

3 In the Storage Providers list, view the storage providers registered with vCenter Server.

   The list shows general information including the name of the storage provider, its URL and status, and version of VASA APIs.

4 To display additional details, click the icon next to the provider's name.

   Note A single storage provider can support storage systems from multiple different vendors.

   You can display information about provider's certificates and storage entities it represents.

Re-authenticate Host VASA Client

Starting from vSphere 8.0 Update 3, you can monitor the storage provider connection and authentication status on each ESXi host. If there is an authentication issue in a specific host, you can re-authenticate the specific host against the storage provider.

Use the vSphere Client to monitor the storage providers.

Prerequisites

Register storage providers. See Register Storage Providers for Virtual Volumes.

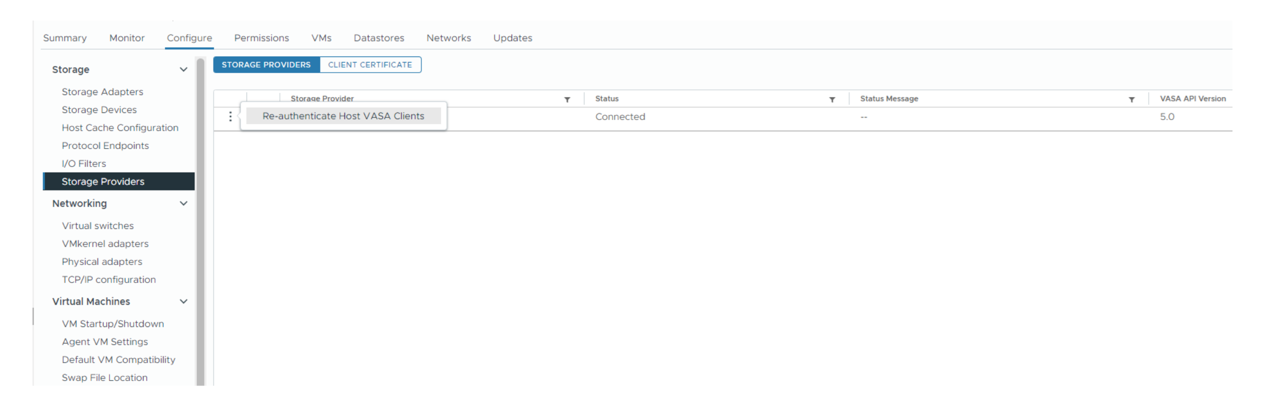
Procedure

1 Navigate to the ESXi host in vCenter Server.

2 Click the **Configure** tab, and click **Storage Providers**.

   The **Storage Providers** tab lists the storage provider registered with the ESXi host. You can view general information including the name of the storage provider, status, and version of the VASA API. The **Client Certificate** lists the VASA client certificate details.

3 If the status indicates an authentication error for a provider, click the row action (three dots) icon next to the provider's name and click **Re-authenticate Host VASA Clients**.



   The VASA client re-authorizes the ESXi host with the storage provider.

Manage Storage Providers

Use the vSphere Client to perform several management operations on the registered storage providers.

Procedure

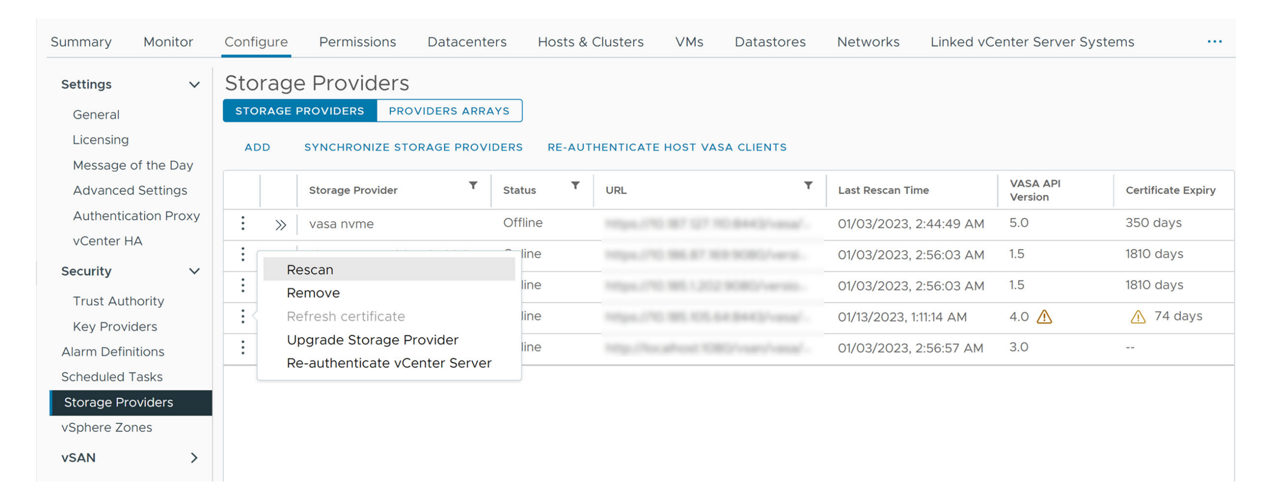
1 Navigate to vCenter Server.

2 Click the **Configure** tab, and click **Storage Providers**.

3 Click the icon to perform one of the following global actions for all providers.

Icon	Description
Synchronize Storage Providers	Synchronize all storage providers with the current state of the environment.
Re-authenticate Host VASA Clients	Re-authenticate all hosts with Virtual Volumes datastores against their respective storage providers. This icon is available only for Virtual Volumes providers. **Note** This operation is time consuming.

4 To manage a specific storage provider, navigate to the provider and click the **Row action** (three dots) icon next to the provider's name.



5 Select one of the following from the **Row actions** menu.

Availability of the menu options depends on the type of the storage provider you use. For example, you cannot manually unregister or upgrade certain storage providers supplied by VMware, such as vSAN or I/O filter storage providers.

Option	Description
Rescan	Update storage data for the provider. vCenter Server periodically updates storage data in its database. The updates are partial and reflect only those changes that storage providers communicate to vCenter Server at that moment. When needed, you can perform a full database synchronization for the selected storage provider.
Remove	Unregister storage providers that you do not use. After this operation, vCenter Server closes the connection and removes the storage provider from its configuration.
Refresh certificate	vCenter Server warns you when a certificate assigned to a storage provider is about to expire. You can refresh the certificate to continue using the provider. If you fail to refresh the certificate before it expires, vCenter Server discontinues using the provider.

Option	Description
Upgrade Storage Provider	When vCenter Server notifies you about a new available VASA version, accept this version to enable it in your vSphere environment. vCenter Server activates new capabilities exposed by the storage provider and communicates this information to all hosts in the inventory. When you perform this action, you must enter credentials of the storage provider. You cannot select specific VASA version for an upgrade if several versions are available. vCenter Server upgrades only to the latest version. **Note** You have the flexibility to upgrade at your own pace. For more information, see vSphere Virtual Volumes Security.
Re-authenticate vCenter Server	Re-authenticate vCenter Server against the storage provider. This action re-authorizes vCenter Server with the storage provider if it has lost authentication and would need credentials. You must enter credentials of the storage provider.

Working with VMware vSphere Virtual Volumes

14

VMware vSphere Virtual Volumes, also known as vVols, virtualizes storage devices by abstracting physical hardware resources into logical pools of capacity. The Virtual Volumes functionality changes the storage management paradigm from managing space inside datastores to managing abstract storage objects handled by storage arrays. With Virtual Volumes, an individual virtual machine, not the datastore, becomes a unit of storage management, while storage hardware gains complete control over virtual disk content, layout, and management.

Virtual Volumes maps virtual disks and their derivatives, clones, snapshots, and replicas, directly to objects, called virtual volumes, on a storage system. This mapping allows vSphere to offload intensive storage operations such as snapshot, cloning, and replication to the storage system.

Read the following topics next:

- vSphere Virtual Volumes Concepts and Architecture
- vSphere Virtual Volumes Security
- vSphere Virtual Volumes Snapshots
- vSphere Virtual Volumes and Shared Virtual Disks
- Before You Enable vSphere Virtual Volumes
- Configure vSphere Virtual Volumes
- Provision VMs on vSphere Virtual Volumes Datastores
- Review and Manage Protocol Endpoints in vSphere
- Manage Storage Providers for vSphere Virtual Volumes
- Re-authenticate Host VASA Client
- NVMe and Virtual Volumes in vSphere
- vSphere Virtual Volumes and Replication
- Best Practices for Working with vSphere Virtual Volumes
- Using Stretched Storage Clustering with Virtual Volumes
- Using the `esxcli storage vvol` Command to Troubleshoot vSphere Virtual Volumes

vSphere Virtual Volumes Concepts and Architecture

Learn about basic concepts and different components of the vSphere Virtual Volumes functionality.

Watch the following video to learn about Virtual Volumes concepts.



(Virtual Volumes Part 1: Concepts)

Virtual Volume Objects

Virtual volumes are encapsulations of virtual machine files, virtual disks, and their derivatives.

Virtual volumes are stored natively inside a storage system that is connected to your ESXi hosts through Ethernet or SAN. They are exported as objects by a storage system and are managed entirely by hardware on the storage side. Typically, a unique GUID identifies a virtual volume. Virtual volumes are not preprovisioned, but created automatically when you perform virtual machine management operations. These operations include a VM creation, cloning, and snapshotting. ESXi and vCenter Server associate one or more virtual volumes to a virtual machine.

Types of Virtual Volumes

The system creates the following types of virtual volumes for the core elements that make up the virtual machine:

Data-vVol

A data virtual volume that corresponds directly to each virtual disk `.vmdk` file. As virtual disk files on traditional datastores, virtual volumes are presented to virtual machines as SCSI or NVMe disks. Data-vVol can be either thick or thin-provisioned.

Config-vVol

A configuration virtual volume, or a home directory, represents a small directory that contains metadata files for a virtual machine. The files include a `.vmx` file, descriptor files for virtual disks, log files, and so forth. The configuration virtual volume is formatted with a file system. When ESXi uses the SCSI or NVMe protocol to connect to storage, configuration virtual volumes are formatted with VMFS. With NFS protocol, configuration virtual volumes are presented as an NFS directory. Typically, it is thin-provisioned.

Starting with vSphere 7.0 Update 2, partners can increase the config-vVol to above 4 GB. Work with your Virtual Volumes partner on implementing this if it is supported by your partner and applicable to your environment.

vSphere 8.0 Update 2 supports space reclamation for config-vVols that reside on Virtual Volumes datastores accessed through the SCSI or NVMe protocols. For more information, see Reclaim Space on the vSphere Virtual Volumes Datastores

Swap-vVol

Created when a VM is first powered on. It is a virtual volume to hold copies of VM memory pages that cannot be retained in memory. Its size is determined by the VM's memory size. It is thick-provisioned by default.

Snapshot-vVol

A virtual memory volume to hold the contents of virtual machine memory for a snapshot. Thick-provisioned.

For more information, see vSphere Virtual Volumes Snapshots.

Other

A virtual volume for specific features. For example, a digest virtual volume is created for Content-Based Read Cache (CBRC).

Typically, a VM creates a minimum of three virtual volumes, data-vVol, config-vVol, and swap-vVol. The maximum depends on how many virtual disks and snapshots reside on the VM.

By using different virtual volumes for different VM components, you can apply and manipulate storage policies at the finest granularity level. For example, a virtual volume that contains a virtual disk can have a richer set of services than the virtual volume for the VM boot disk.

Disk Provisioning

The Virtual Volumes functionality supports a concept of thin and thick-provisioned virtual disks. However, from the I/O prospective, implementation and management of thin or thick provisioning by the arrays is transparent to the ESXi host. ESXi offloads to the storage arrays any functions related to thin provisioning.

You select the thin or thick type for your virtual disk at the VM creation time. If your disk is thin and resides on a Virtual Volumes datastore, you cannot change its type later by inflating the disk.

Shared Disks

You can place a shared disk on a Virtual Volumes storage that supports SCSI Persistent Reservations for Virtual Volumes. You can use this disk as a quorum disk and eliminate RDMs in the MSCS clusters. For more information, see the *vSphere Resource Management* documentation.

Virtual Volumes Storage Providers

A Virtual Volumes storage provider, also called a VASA provider, is a software component that acts as a storage awareness service for vSphere. The provider mediates out-of-band communication between vCenter Server and ESXi hosts on one side and a storage system on the other.

The storage provider is implemented through VMware APIs for Storage Awareness (VASA) and is used to manage all aspects of Virtual Volumes storage.

The storage provider delivers information from the underlying storage container. The storage container capabilities appear in vCenter Server and the vSphere Client. Then, in turn, the storage provider communicates virtual machine storage requirements, which you can define in the form of a storage policy, to the storage layer. This integration process ensures that a virtual volume created in the storage layer meets the requirements outlined in the policy.

Typically, vendors are responsible for supplying storage providers that can integrate with vSphere and provide support to Virtual Volumes. Every storage provider must be certified by VMware and properly deployed. For information about deploying and upgrading the Virtual Volumes storage provider to a version compatible with current ESXi release, contact your storage vendor.

After you deploy the storage provider, you must register it in vCenter Server. See Register Storage Providers for Virtual Volumes. To upgrade your storage providers or for other operations that you can perform, see Manage Storage Providers for vSphere Virtual Volumes.

Virtual Volumes Storage Containers

Unlike traditional block or file based storage, the Virtual Volumes functionality does not require preconfigured storage on a storage side. Instead, Virtual Volumes uses a storage container. It is a pool of raw storage capacity or an aggregation of storage capabilities that a storage system can provide to virtual volumes.

A storage container is a part of the logical storage fabric and is a logical unit of the underlying hardware. The storage container logically groups virtual volumes based on management and administrative needs. For example, the storage container can contain all virtual volumes created for a tenant in a multitenant deployment, or a department in an enterprise deployment. Each storage container serves as a virtual volume store and virtual volumes are allocated out of the storage container capacity.

Typically, a storage administrator on the storage side defines storage containers. The number of storage containers, their capacity, and their size depend on a vendor-specific implementation. At least one container for each storage system is required.

Note A single storage container cannot span different physical arrays.

After you register a storage provider associated with the storage system, vCenter Server discovers all configured storage containers along with their storage capability profiles, protocol endpoints, and other attributes. A single storage container can export multiple capability profiles. As a result, virtual machines with diverse needs and different storage policy settings can be a part of the same storage container.

Initially, all discovered storage containers are not connected to any specific host, and you cannot see them in the vSphere Client. To mount a storage container, you must map it to a Virtual Volumes datastore.

Static Protocol Endpoints

With SCSI or NFS transports, ESXi hosts use a logical I/O proxy, called the protocol endpoint, to communicate with virtual volumes. ESXi uses protocol endpoints to establish a data path on demand from virtual machines to their respective virtual volumes.

Note Information in this section applies only to static protocol endpoints that use the SCSI or NFS transports. For specifics about the NVMe protocol endpoints, see NVMe and Virtual Volumes in vSphere.

Each virtual volume is bound to a specific protocol endpoint. When a virtual machine on the host performs an I/O operation, the protocol endpoint directs the I/O to the appropriate virtual volume. Typically, a storage system requires just a few protocol endpoints. A single protocol endpoint can connect to hundreds or thousands of virtual volumes.

On the storage side, a storage administrator configures protocol endpoints, one or several per storage container. The protocol endpoints are a part of the physical storage fabric. The storage system exports the protocol endpoints with associated storage containers through the storage provider. After you map the storage container to a Virtual Volumes datastore, the ESXi host discovers the protocol endpoints and they become visible in the vSphere Client. The protocol endpoints can also be discovered during a storage rescan. Multiple hosts can discover and mount the protocol endpoints.

In the vSphere Client, the list of available protocol endpoints looks similar to the host storage devices list. Different storage transports can be used to expose the protocol endpoints to ESXi. When the SCSI-based transport is used, the protocol endpoint represents a proxy LUN defined by a T10-based LUN WWN. For the NFS protocol, the protocol endpoint is a mount point, such as an IP address and a share name. You can configure multipathing on the SCSI-based protocol endpoint, but not on the NFS-based protocol endpoint. No matter which protocol you use, the storage array can provide multiple protocol endpoints for availability purposes.

Protocol endpoints are managed per array. ESXi and vCenter Server assume that all protocol endpoints reported for an array are associated with all containers on that array. For example, if an array has two containers and three protocol endpoints, ESXi assumes that virtual volumes on both containers can be bound to all three protocol points.

For information about viewing the static protocol endpoints in the vSphere Client, see Review and Manage Static Protocol Endpoints.

Binding and Unbinding Virtual Volumes

At the time of creation, a virtual volume is a passive entity and is not immediately ready for I/O. To access the virtual volume, ESXi or vCenter Server send a bind request.

The storage system replies with a protocol endpoint ID that becomes an access point to the virtual volume. The protocol endpoint accepts all I/O requests to the virtual volume. This binding exists until ESXi sends an unbind request for the virtual volume.

For later bind requests on the same virtual volume, the storage system can return different protocol endpoint IDs.

When you use the NVMe protocol, the bind virtual volume response provides the NVMe subsytem NQN and the namespace ID (nsid) of the namespace virtual volume object. The ESXi host uses this information and resolves this to the ANA group within the subsystem. Corresponding to this ANA group, a virtual protocol endpoint (vPE) is created if it does not exist. It is used to direct all I/O requests to Virtual Volumes.

When receiving concurrent bind requests to a virtual volume from multiple ESXi hosts, the storage system can return the same or different endpoint bindings to each requesting ESXi host. In other words, the storage system can bind different concurrent hosts to the same virtual volume through different endpoints.

The unbind operation removes the I/O access point for the virtual volume. The storage system might unbind the virtual volume from its protocol endpoint immediately, or after a delay, or take some other action. A bound virtual volume cannot be deleted until it is unbound.

Virtual Volumes Datastores

A Virtual Volumes datastore represents a storage container in vCenter Server and the vSphere Client.

After vCenter Server discovers storage containers exported by storage systems, you must mount them as Virtual Volumes datastores. The Virtual Volumes datastores are not formatted in a traditional way like, for example, VMFS datastores. You must still create them because all vSphere functionalities, including FT, HA, DRS, and so on, require the datastore construct to function properly.

You use the datastore creation wizard in the vSphere Client to map a storage container to a Virtual Volumes datastore. The Virtual Volumes datastore that you create corresponds directly to the specific storage container.

From a vSphere administrator prospective, the Virtual Volumes datastore is similar to any other datastore and is used to hold virtual machines. Like other datastores, the Virtual Volumes datastore can be browsed and lists virtual volumes by virtual machine name. Like traditional datastores, the Virtual Volumes datastore supports unmounting and mounting. However, such operations as upgrade and resize are not applicable to the Virtual Volumes datastore. The Virtual Volumes datastore capacity is configurable by the storage administrator outside of vSphere.

You can use the Virtual Volumes datastores with traditional VMFS and NFS datastores and with vSAN.

Note The size of a virtual volume must be a multiple of 1 MB, with a minimum size of 1 MB. As a result, all virtual disks that you provision on a Virtual Volumes datastore must be an even multiple of 1 MB. If the virtual disk you migrate to the Virtual Volumes datastore is not an even multiple of 1 MB, extend the disk to the nearest even multiple of 1 MB.

To create a Virtual Volumes datastore, see Create a Virtual Volumes Datastore in vSphere Environment.

Virtual Volumes and VM Storage Policies

A virtual machine that runs on a Virtual Volumes datastore requires a VM storage policy.

A VM storage policy is a set of rules that contains placement and quality-of-service requirements for a virtual machine. The policy enforces appropriate placement of the virtual machine within Virtual Volumes storage and guarantees that storage can satisfy virtual machine requirements.

You use the VM Storage Policies interface to create a Virtual Volumes storage policy. When you assign the new policy to the virtual machine, the policy enforces that the Virtual Volumes storage meets the requirements.

Virtual Volumes Default Storage Policy

For Virtual Volumes, VMware provides a default storage policy that contains no rules or storage requirements, called Virtual Volumes No Requirements Policy. This policy is applied to the VM objects when you do not specify another policy for the virtual machine on the Virtual Volumes datastore. With the No Requirements policy, storage arrays can determine the optimum placement for the VM objects.

The default No Requirements policy that VMware provides has the following characteristics:

- You cannot delete, edit, or clone this policy.
- The policy is compatible only with the Virtual Volumes datastores.
- You can create a VM storage policy for Virtual Volumes and designate it as the default.

Virtual Volumes and Storage Protocols

A Virtual Volumes storage system provides protocol endpoints that are discoverable on the physical storage fabric. ESXi hosts use the protocol endpoints to connect to virtual volumes on the storage. Operation of the protocol endpoints depends on storage protocols that expose the endpoints to ESXi hosts.

Virtual Volumes supports NFS version 3 and 4.1, iSCSI, Fibre Channel, FCoE, NVMe over Fibre Channel, and NVMe over TCP.

Irrespective of which storage protocol is used, protocol endpoints provide uniform access to both SAN and NAS storage. A virtual volume, like a file on other traditional datastore, is presented to a virtual machine as a SCSI or an NVMe disk.

Note A storage container is dedicated to SAN (SCSI or NVMe) or NAS and cannot be shared across those protocol types. An array can present one storage container with SCSI protocol endpoints and a different container with NFS protocol endpoints. The container cannot use a combination of SCSI, NVMe, and NFS storage access protocols.

Virtual Volumes and SCSI-Based Transports

On disk arrays, Virtual Volumes supports Fibre Channel, FCoE, and iSCSI protocols.

When the SCSI-based protocol is used, the protocol endpoint represents a proxy LUN defined by a T10-based LUN WWN.

As any block-based LUNs, the protocol endpoints are discovered using standard LUN discovery commands. The ESXi host periodically rescans for new devices and asynchronously discovers block-based protocol endpoints. The protocol endpoint can be accessible by multiple paths. Traffic on these paths follows well-known path selection policies, as is typical for LUNs.

On SCSI-based disk arrays at VM creation time, ESXi makes a virtual volume and formats it as VMFS. This small virtual volume stores all VM metadata files and is called the config-vVol. The config-vVol functions as a VM storage locator for vSphere.

Virtual Volumes on disk arrays supports the same set of SCSI commands as VMFS and use ATS as a locking mechanism.

CHAP Support for iSCSI Endpoints

Virtual Volumes supports Challenge Handshake Access Protocol (CHAP) with iSCSI targets. This support allows ESXi hosts to share CHAP initiator credentials with Virtual Volumes storage providers, also called VASA providers, and for Virtual Volumes storage providers to raise system events notifying vCenter Server of changes to CHAP target credentials on the storage array.

Each ESXi host can have multiple HBAs and each HBA can have properties configured on it. One of these properties is the authentication method that the HBA must use. Authentication is optional, but if implemented, it must be supported by both the initiator and target. CHAP is an authentication method that can be used both directions between initiator and target.

For more information about different CHAP authentication methods, see Selecting CHAP Authentication Method. To configure CHAP on your ESXi host, see Configuring CHAP Parameters for iSCSI or iSER Storage Adapters on ESXi Host.

Virtual Volumes and NFS Transports

With NAS storage, a protocol endpoint is an NFS share that the ESXi host mounts using IP address or DNS name and a share name. Virtual Volumes supports NFS version 3 and 4.1 to access NAS storage. Both IPv4 and IPv6 formats are supported.

No matter which version you use, a storage array can provide multiple protocol endpoints for availability purposes.

In addition, NFS version 4.1 introduces trunking mechanisms that enable load balancing and multipathing.

Virtual Volumes on NAS devices supports the same NFS Remote Procedure Calls (RPCs) that ESXi hosts use when connecting to NFS mount points.

On NAS devices, a config-vVol is a directory subtree that corresponds to a config-vVolID. The config-vVol must support directories and other operations that are necessary for NFS.

Virtual Volumes and NVMe

Virtual Volumes supports NVMe protocols, including NVMe over Fibre Channel and NVMe over TCP. A virtual volume object is mapped to a namespace within an NVMe subsystem. ANA groups within the NVMe subsystem are viewed as virtual protocol endpoints on the ESXi host.

Virtual protocol endpoints are used for path state management as the ANA group state changes. The ESXi host discovers the ANA groups dynamically, as and when required. This means that the virtual protocol endpoint is created only when the ESXi host needs I/O access to a namespace virtual volume within the NVMe subsystem. Config-vVols on NVMe are similar to SCSI that are formatted with VMFS. They are also used to store the VM metadata files.

To configure NVMe with Virtual Volumes on your ESXi host, see NVMe and Virtual Volumes in vSphere.

Virtual Volumes Architecture

An architectural diagram provides an overview of how all components of the Virtual Volumes functionality interact with each other.

Watch the video for information about Virtual Volumes architecture.



(Virtual Volumes Part 2: Architecture)

vSphere Virtual Volumes Security

Learn about security improvements that vSphere Virtual Volumes storage provider, also called VASA provider, version 5 offers in vSphere 8.0 Update 1 and later releases. You can also learn about the security model in vSphere 8.0 and earlier releases.

VASA 5 Support and Security with vSphere 8.0 Update 1 and Later

All storage providers of VASA 5 and later versions use a stricter authentication mechanism, which requires ESXi to be authenticated in context of vCenter Server. VASA 5 improves security and offers a significantly modified management model that includes the following major changes:

- For each vCenter Server that registers with the array using VASA 5 or later, the VASA provider creates a dedicated web server instance, or a virtual host, which can be either a virtual web server instance or a completely separate instance. The VASA client in vCenter Server relies on certificate based authentication and authorization to access its dedicated virtual host created on the array. All VASA client certificates, including vCenter Server and ESXi certificates, get registered with the virtual host. This creates strong isolation between different vCenter Server systems when the systems are authenticated. In addition, VASA providers can offer separate resource access and improved isolation to different vCenter Server systems.
- With VASA 5, the VASA client uses a dedicated certificate for VASA communications. Each vCenter Server provisions a certificate for the VASA provider, which is managed through a dedicated virtual host specific to vCenter Server. All ESXi hosts that support VASA 5 use the dedicated virtual host created by their managing vCenter Server.
- vCenter Server provisions the VASA client certificate for each new ESXi host 8.0 Update 1 or later and synchronizes the public key of the certificate with the VASA provider. In contrast with the previous security model that authenticated the CA issuer for the client certificate, the VASA provider now identifies and authorizes an individual client using the public key.
- To comply with VMware security requirements, vSphere does not trust self-signed certificates for TLS communications. The only exception is during a short period when the VASA provider gets registered and for backward compatibility. An array administrator can use a Custom CA certificate for the VASA provider to override the self-signed certificate at array for backward compatibility and bootstrapping.
- To avoid losing access to the VASA provider, follow these guidelines. For information, see Manage Storage Providers for vSphere Virtual Volumes.
  - Do not unregister and re-register your VASA provider to upgrade. Instead, use a proper upgrade mechanism for your VASA provider. When vCenter Server notifies you about a new available VASA version, make sure to accept this version within a reasonable time. To upgrade from the vSphere Client, use the **Upgrade Storage Provider** option.
  - Regularly refresh the VASA provider certificate. Make sure to refresh the certificate within a reasonable time after vCenter Server warns you that the certificate assigned to the VASA provider is about to expire. Use the vSphere Client **Refresh certificate** option.
  - When you renew the VMCA root certificate or vCenter Server client certificate, SMS might lose connection with the VASA provider. If the provider is offline, use the **Re-authenticate vCenter Server** option.
  - If a host loses authentication, you can remediate the authentication failure by using the **Re-authenticate Host VASA Clients** option.

Security Certificates with vSphere 8.0 and Earlier Releases

vSphere includes the VMware Certificate Authority (VMCA). By default, the VMCA creates all internal certificates used in vSphere environment. It generates certificates for newly added ESXi hosts and storage VASA providers that manage or represent Virtual Volumes storage systems.

Communication with the VASA provider is protected by SSL certificates. These certificates can come from the VASA provider or from the VMCA.

- Certificates can be directly provided by the VASA provider for long-term use. They can be either self-generated and self-signed, or derived from an external Certificate Authority.
- Certificates can be generated by the VMCA for use by the VASA provider.

When a host or VASA provider is registered, VMCA follows these steps automatically, without involvement from the vSphere administrator.

1. When a VASA provider is first added to the vCenter Server storage management service (SMS), it produces a self-signed certificate.
2. After verifying the certificate, the SMS requests a Certificate Signing Request (CSR) from the VASA provider.
3. After receiving and validating the CSR, the SMS presents it to the VMCA on behalf of the VASA provider, requesting a CA signed certificate.

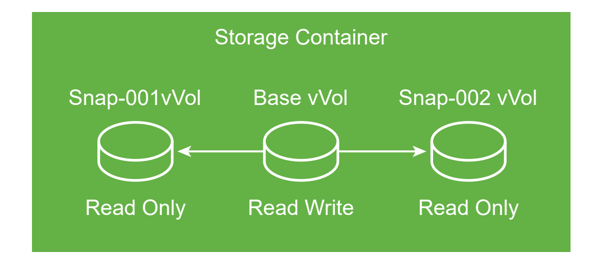
   The VMCA can be configured to function as a standalone CA, or as a subordinate to an enterprise CA. If you set up the VMCA as a subordinate CA, the VMCA signs the CSR with the full chain.
4. The signed certificate with the root certificates is passed to the VASA provider. The VASA provider can authenticate all future secure connections originating from the SMS on vCenter Server and on ESXi hosts.

vSphere Virtual Volumes Snapshots

In vSphere, snapshots preserve the state and data of a virtual machine at the time you take the snapshot. Snapshots are useful when you must revert repeatedly to the same virtual machine state, but you do not want to create multiple virtual machines. Virtual Volumes snapshots serve many purposes. You can use them to create a quiesced copy for backup or archival purposes, or to create a test and rollback environment for applications. You can also use them to provision application images instantly.

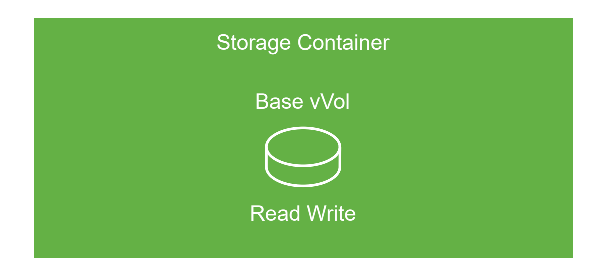
In Virtual Volumes environment, snapshots are managed by ESXi and vCenter Server, but are performed by the storage array.

Each snapshot creates an extra virtual volume object, a snapshot virtual volume, that holds the contents of virtual machine memory. Original VM data is copied to this object, and it remains read-only, which prevents the guest operating system from writing to snapshot. You cannot resize the snapshot virtual volume. Typically, when you replicate the VM, its snapshot virtual volume is also replicated.



The base virtual volume remains active, or read-write. When another snapshot is created, it preserves the new state and data of the virtual machine at the time you take the snapshot.

When you delete snapshots, only the base virtual volume remains, while snapshot virtual volume objects are discarded. The base virtual volume represents the most current state of the virtual machine. Unlike snapshots on the traditional datastores, snapshot virtual volumes do not need to commit their contents to the base virtual volume.



For information about creating and managing snapshots, see the *vSphere Virtual Machine Administration* documentation.

vSphere Virtual Volumes and Shared Virtual Disks

Virtual Volumes support shared virtual disks that can be used with such clustering software as Windows Server Failover Clustering (WSFC). Starting with vSphere 8.0 Update 2, Virtual Volumes support the hot extension functionality for shared virtual disks on the Virtual Volumes datastore. With this functionality, you can increase the size of a shared disk without bringing the cluster down.

To learn more about the support Virtual Volumes provide to shared disks with WSFC, see the following information:

- VMware vSphere® Virtual Volumes Support for WSFC
- Virtual Volumes (vVols) now supports WSFC
- vVols and Microsoft WSFC

For information about setting up and extending the shared virtual disk on the Virtual Volumes datastore, see Add Hard Disks to the First Node of a WSFC Across Physical Hosts with vVols.

Before You Enable vSphere Virtual Volumes

To work with vSphere Virtual Volumes, ensure that your storage and vSphere environment are set up correctly.

Prepare Storage System for Virtual Volumes

To prepare your storage system environment for Virtual Volumes, follow these guidelines. For additional information, contact your storage vendor.

- The storage system or storage array that you use must support Virtual Volumes and integrate with the vSphere components through vSphere APIs for Storage Awareness (VASA). The storage array must support thin provisioning and snapshotting.
- The Virtual Volumes storage provider must be deployed.
- The following components must be configured on the storage side:
  - Protocol endpoints. Except NVMe.
  - Storage containers
  - Storage profiles
  - Replication configurations if you plan to use Virtual Volumes with replication. See Requirements for Replication with Virtual Volumes.
- If you use NVMe, create NVM subsystems and associate Virtual Volumes Host NQNs with relevant NVM subsystems. This configuration step depends on partner specific NVMe design. For more information, contact your array vendor.

Prepare vSphere Environment

- Ensure to follow appropriate setup guidelines for the type of storage you use, NVMe over Fibre Channel, NVMe over TCP, Fibre Channel, FCoE, iSCSI, or NFS. If necessary, install and configure storage adapters on your ESXi hosts.
  - If you use iSCSI, activate the software iSCSI adapters on your ESXi hosts. Configure Dynamic Discovery and enter the IP address of your Virtual Volumes storage system. See Configure the Software iSCSI Adapter with ESXi .
  - If you use NVMe, follow the requirements for NVMe over Fibre Channel and NVMe over TCP. See Requirements for VMware NVMe Storage.
- Synchronize all components in the storage array with vCenter Server and all ESXi hosts. Use Network Time Protocol (NTP) to do this synchronization.

For more information, contact your vendor and see *VMware Compatibility Guide*

Synchronize vSphere Storage Environment with a Network Time Server

If you use Virtual Volumes, configure Network Time Protocol (NTP) to make sure all ESXi hosts on the vSphere network are synchronized.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, select **Time Configuration**.

4 Click **Edit** and set up the NTP server.

    a Select **Use Network Time Protocol (Enable NTP client)**.

    b Set the NTP Service Startup Policy.

    c Enter the IP addresses of the NTP server to synchronize with.

    d Click **Start** or **Restart** in the NTP Service Status section.

5 Click **OK**.

    The host synchronizes with the NTP server.

Configure vSphere Virtual Volumes

To configure Virtual Volumes in the vSphere environment, follow several steps.

Prerequisites

- Be familiar with concepts and requirements related to Virtual Volumes in the vSphere environment. See vSphere Virtual Volumes Concepts and Architecture and NVMe and Virtual Volumes in vSphere.
- Follow guidelines in Before You Enable vSphere Virtual Volumes.

Procedure

1 Register Storage Providers for Virtual Volumes.

2 Create a Virtual Volumes Datastore in vSphere Environment.

Register Storage Providers for Virtual Volumes

Your Virtual Volumes environment must include storage providers, also called VASA providers. Typically, third-party vendors develop storage providers through the VMware APIs for Storage Awareness (VASA). Storage providers facilitate communication between vSphere and the storage side. Use the vSphere Client to register the Virtual Volumes storage providers.

After registration, the Virtual Volumes provider communicates with vCenter Server. The provider reports characteristics of underlying storage and data services, such as replication, that the storage system provides. The characteristics appear in the VM Storage Policies interface and can be used to create a VM storage policy compatible with the Virtual Volumes datastore. After you apply this storage policy to a virtual machine, the policy is pushed to Virtual Volumes storage. The policy enforces optimal placement of the virtual machine within Virtual Volumes storage and guarantees that storage can satisfy virtual machine requirements. If your storage provides extra services, such as caching or replication, the policy enables these services for the virtual machine.

For more information about the storage providers, see Virtual Volumes Storage Providers.

Prerequisites

Verify that an appropriate version of the Virtual Volumes storage provider is installed on the storage side. Obtain credentials of the storage provider.

Procedure

1 Navigate to vCenter Server.

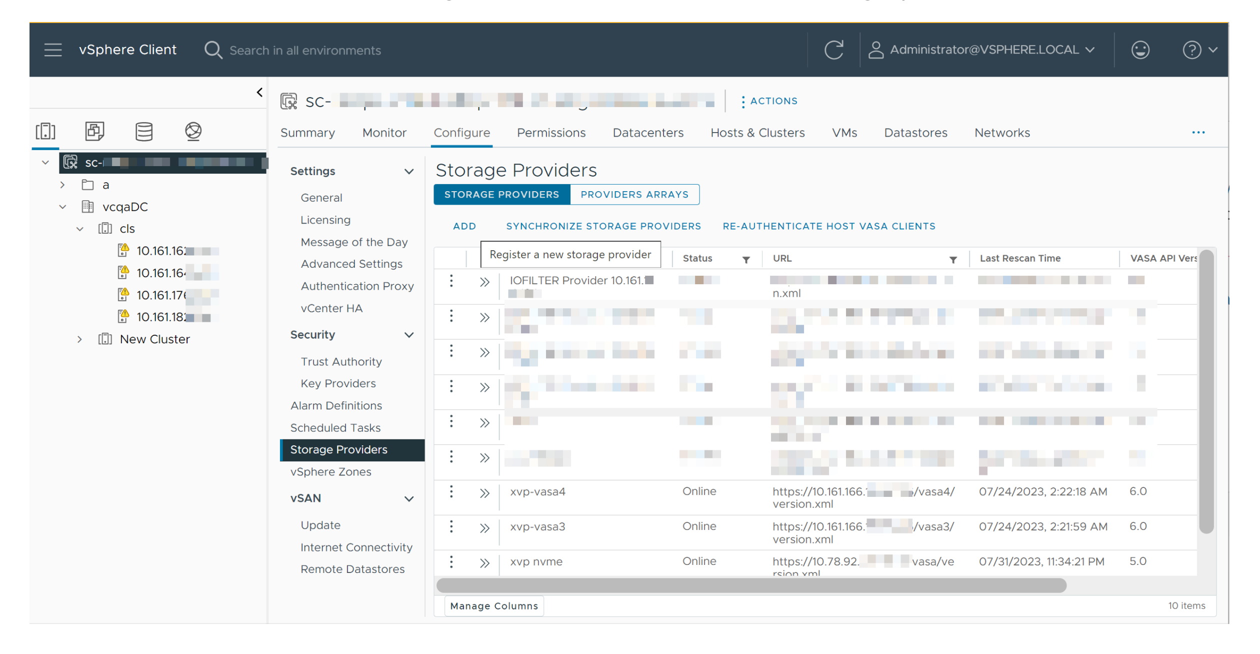
2 Click the **Configure** tab, and click **Storage Providers**.

3 Click the **Add** icon.

4 Enter connection information for the storage provider, including the name, URL, and credentials.

5 To complete the registration, click **OK**.

vCenter Server discovers and registers the Virtual Volumes storage provider.



6 Verify information available for your provider.

Switch between **Storage Providers** and **Providers Arrays** to display information about your provider and the array it represents.

Create a Virtual Volumes Datastore in vSphere Environment

You use the **New Datastore** wizard to create a Virtual Volumes datastore.

For more information about the Virtual Volumes datastore, see Virtual Volumes Datastores.

Procedure

1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.

2 From the right-click menu, select **Storage > New Datastore**.

3 Select **vVol** as the datastore type.

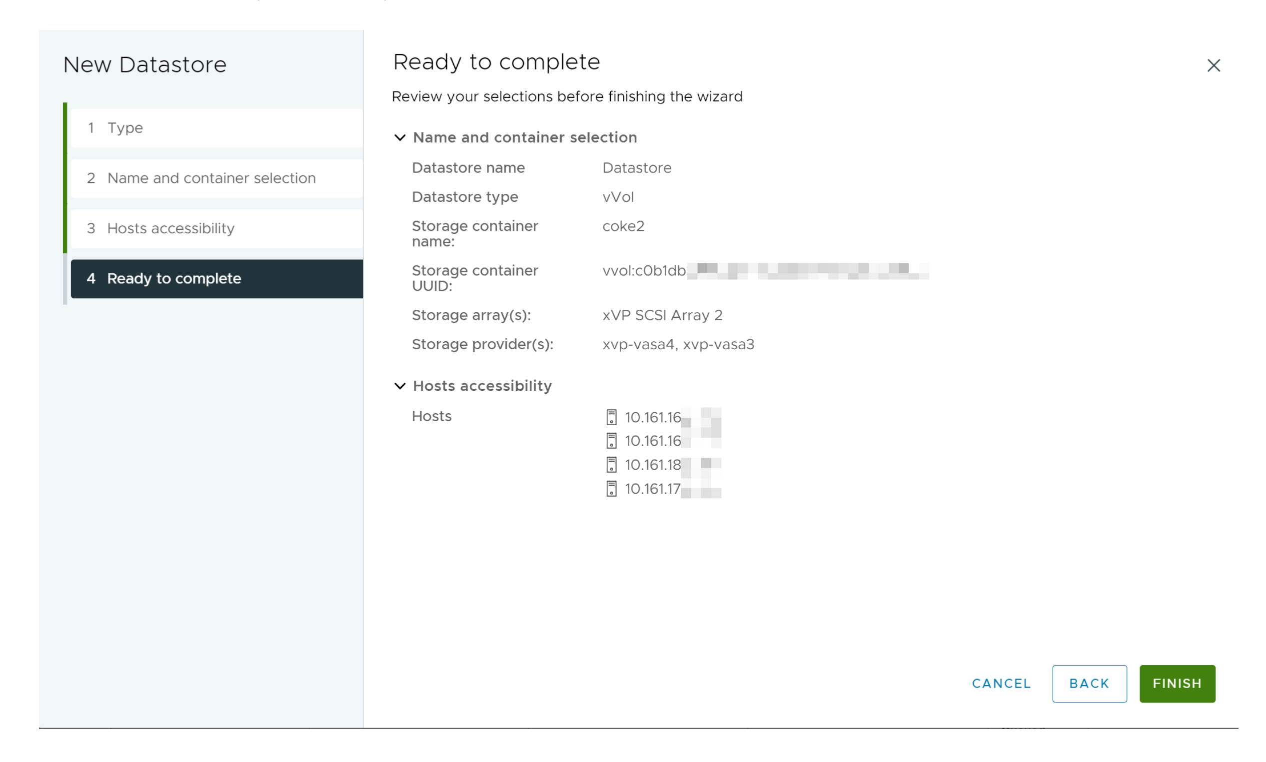
4 Enter the datastore name and select a backing storage container from the list of storage containers.

   Make sure to use the name that does not duplicate another datastore name in your data center environment.

   If you mount the same Virtual Volumes datastore to several hosts, the name of the datastore must be consistent across all hosts.

5 Select the hosts that require access to the datastore.

6 Review the configuration options and click **Finish**.



What to do next

After you create the Virtual Volumes datastore, you can perform the following operations:

- Rename the datastore, browse datastore files, unmount the datastore, and so on. See Basic Administrative Operations for vSphere Datastores.

  Note You cannot add the Virtual Volumes datastore to a datastore cluster.

- Provision virtual machines on the Virtual Volumes datastore. See Provision VMs on vSphere Virtual Volumes Datastores and the *vSphere Virtual Machine Administration* documentation.
- Review and manage protocol endpoints. See Review and Manage Protocol Endpoints in vSphere.
- Manage or upgrade storage providers. See Manage Storage Providers for vSphere Virtual Volumes.

Provision VMs on vSphere Virtual Volumes Datastores

Follow this procedure to provision virtual machines on a vSphere Virtual Volumes datastore.

Note All virtual disks that you provision on a Virtual Volumes datastore must be an even multiple of 1 MB.

A virtual machine that runs on a Virtual Volumes datastore requires an appropriate VM storage policy.

After you provision the virtual machine, you can perform typical VM management tasks. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

1 Define a VM storage policy for Virtual Volumes.

   VMware provides a default No Requirements storage policy for Virtual Volumes. If you need, you can create a custom storage policy compatible with Virtual Volumes.

   See Create a VM Storage Policy for Virtual Volumes.

2 Assign the Virtual Volumes storage policy to the virtual machine.

   To guarantee that the Virtual Volumes datastore fulfills specific storage requirements when allocating a virtual machine, associate the Virtual Volumes storage policy with the virtual machine.

   See Assign Storage Policies to Virtual Machines.

3 Change default storage policy for a Virtual Volumes datastore.

   For virtual machines provisioned on Virtual Volumes datastores, VMware provides a default No Requirements policy. You cannot edit this policy, but you can designate a newly created policy as default.

See Change the Default Storage Policy for a Datastore.

Review and Manage Protocol Endpoints in vSphere

Learn how you can review and manage protocol endpoints that the ESXi host uses to access Virtual Volumes on storage arrays. How the protocol endpoints are represented in the vSphere Client depends on whether the host uses the SCSI and NFS protocols or the NVMe protocol to access Virtual Volumes.

Review and Manage Static Protocol Endpoints

With the SCSI or NFS protocols, the protocol endpoints represent a proxy LUN or an NFS mount point. These protocol points are called static. After you map a storage container to a Virtual Volumes datastore, these protocol endpoints become visible in the vSphere Client. The list of available static protocol endpoints looks similar to the host storage devices list. You can review properties of the protocol endpoints and modify specific settings.

For information about static protocol endpoints that use SCSI or NFS transports, see Static Protocol Endpoints.

Procedure

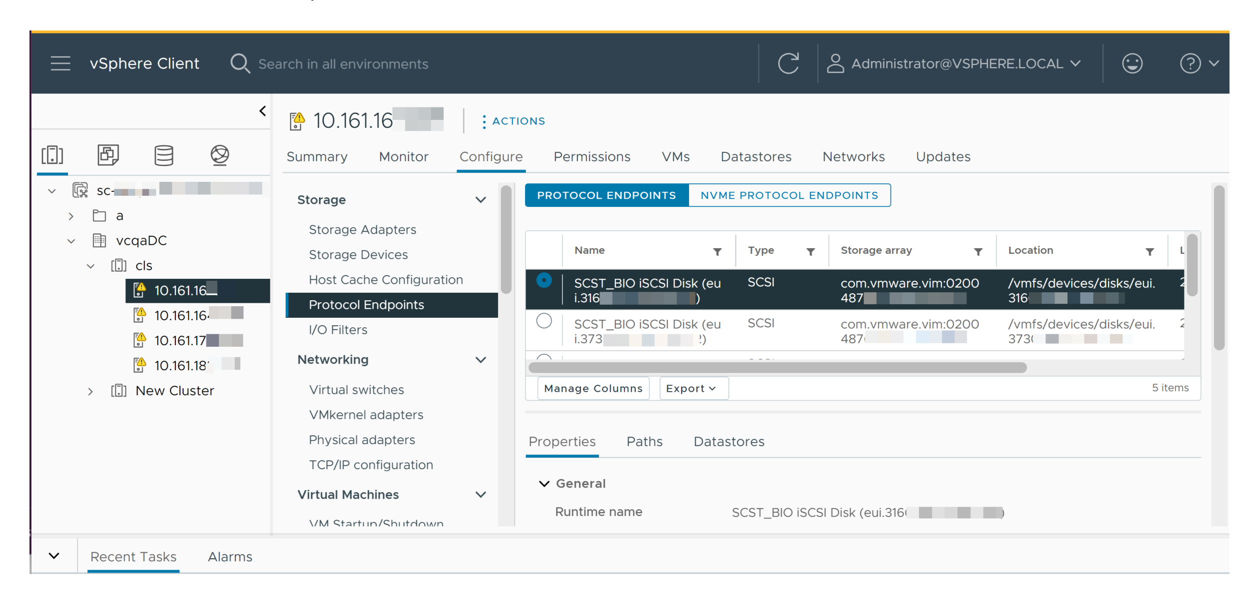
1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Protocol Endpoints**.

4 Click the **Protocol Endpoints** tab.

5 To view details for a specific item, select this item from the list.



6 Use tabs to access additional information and modify properties for the selected protocol endpoint.

Tab	Description
Properties	View the item properties and characteristics. For SCSI (block) items, view and edit multipathing policies.
Paths (SCSI protocol endpoints only)	Display paths available for the protocol endpoint. Deactivate or activate a selected path. Change the Path Selection Policy.
Datastores	Display a corresponding Virtual Volumes datastore. Perform datastore management operations.

Change the Path Selection Policy for a SCSI Protocol Endpoint

If your ESXi host uses SCSI-based transport to communicate with protocol endpoints representing a storage array, you can modify default multipathing policies assigned to protocol endpoints. Use the **Edit Multipathing Policies** dialog box to change a path selection policy.

Procedure

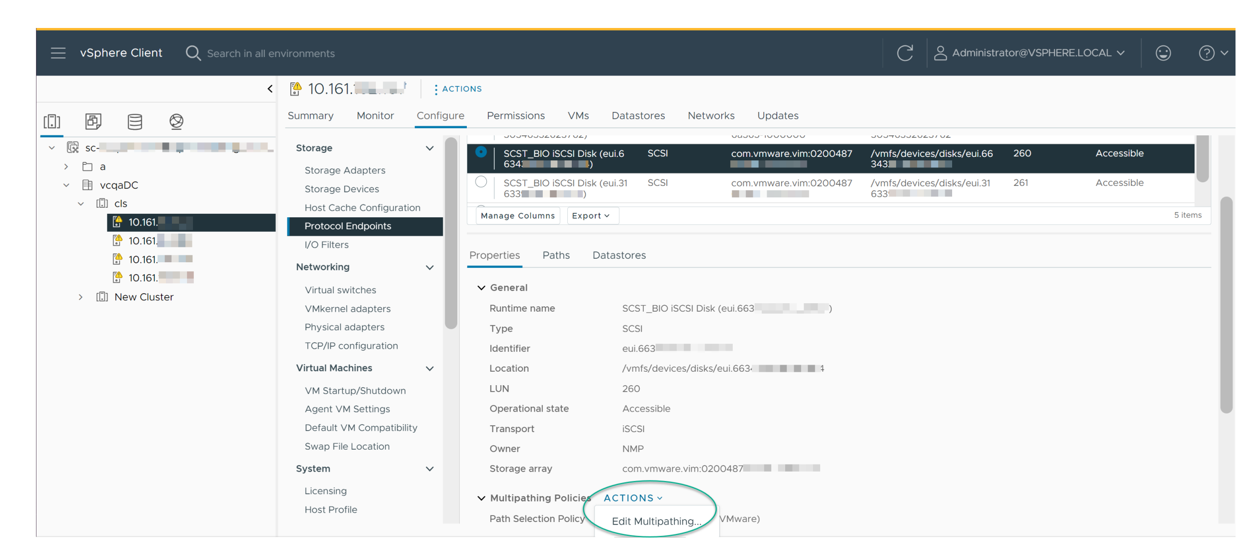
1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Protocol Endpoints**.

4 Select the protocol endpoint whose paths you want to change and click the **Properties** tab.

5 Under Multipathing Policies, select **Edit Multipathing** from the **Actions** menu.



6 Select a path policy and configure its settings.

For information about path policies for SCSI devices, see VMware Path Selection Plug-Ins and Policies.

7 To save your settings and exit the dialog box, click **OK**.

Review NVMe Protocol Endpoints

Unlike the static protocol endpoint, the NVMe protocol endpoint is virtual. This protocol endpoint is a logical representation of the ANA group where the virtual volumes reside. Until a VM is powered on, the NVMe protocol point doesn't exist. Once the VM is powered on, the NVMe protocol point is created, so that the host can access the virtual volumes in the ANA group. Use the vSphere Client to review NVMe virtual protocol endpoints.

For conceptual information that applies to Virtual Volumes with NVMe storage, see NVMe and Virtual Volumes in vSphere.

Procedure

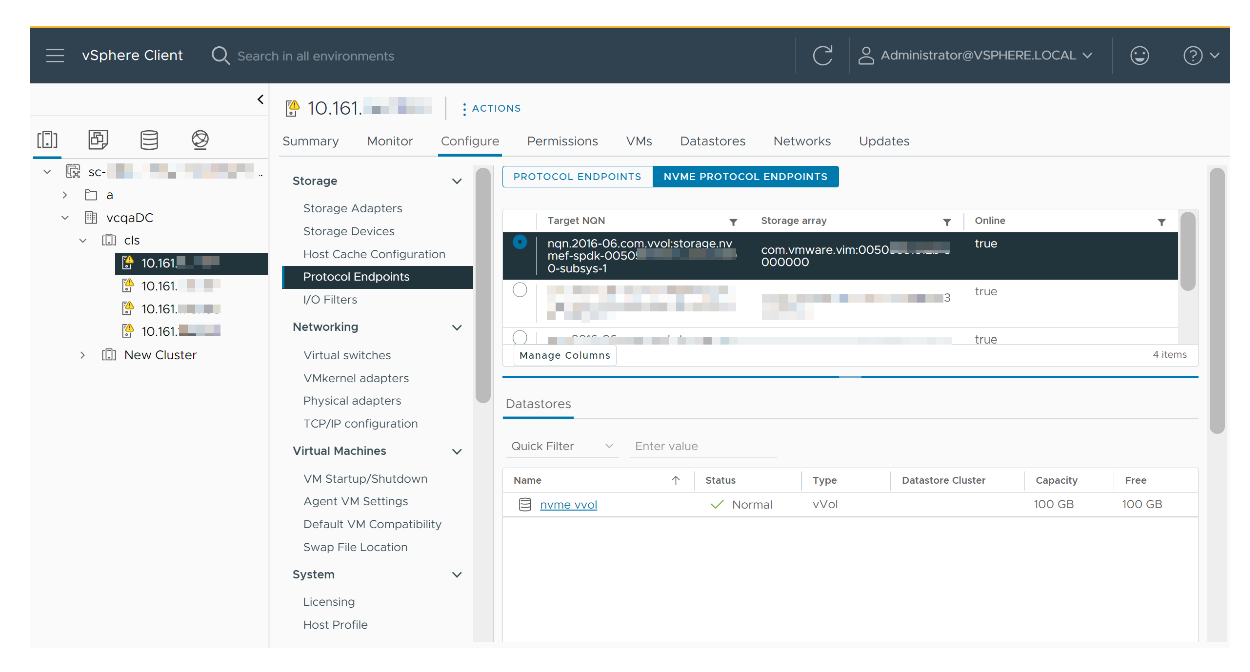
1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **Storage**, click **Protocol Endpoints**.

4 Click the **NVMe Protocol Endpoints** tab and review the list of available NVMe protocol endpoints and their information.

   You can see such details as target NVMe subsystems that are tracked for the NVMe Virtual Volumes datastore.



5 To display additional details, select a protocol endpoint from the list.

   You can display a Virtual Volumes datastore that corresponds to the specific protocol endpoint.

Manage Storage Providers for vSphere Virtual Volumes

After you register storage providers for Virtual Volumes, use the vSphere Client to manage the providers.

For more information about the storage providers, see Virtual Volumes Storage Providers.

Prerequisites

Register storage providers. See Register Storage Providers for Virtual Volumes.

Procedure

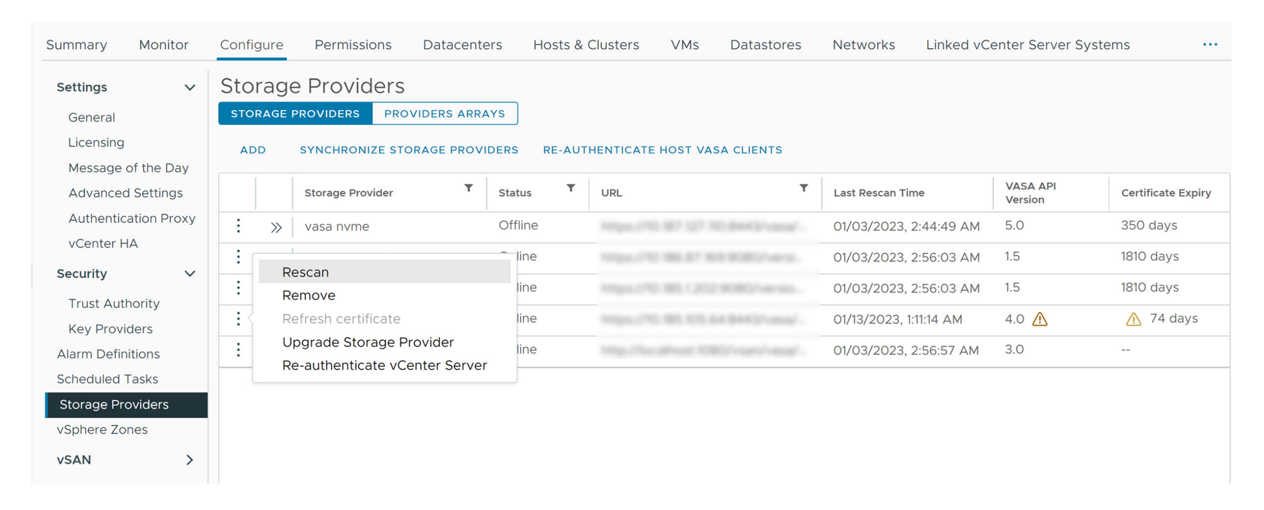
1 Navigate to vCenter Server.

2 Click the **Configure** tab, and click **Storage Providers**.

3 Click the icon to perform one of the following global actions for all Virtual Volumes providers.

Icon	Description
Synchronize Storage Providers	Synchronize all storage providers with the current state of the environment.
Re-authenticate Host VASA Clients	Re-authenticate all hosts against their respective storage providers. Applies to ESXi hosts version 8.0 Update 1 or later. The hosts must support VASA 5. **Note** This operation is time consuming. To re-authenticate only specific hosts, see Re-authenticate Host VASA Client.

4 To manage a specific storage provider, navigate to the provider and click the **Row action** (three dots) icon next to the provider's name.



5 Select one of the following from the **Row actions** menu.

Option	Description
Rescan	Update storage data for the provider. vCenter Server periodically updates storage data in its database. The updates are partial and reflect only those changes that storage providers communicate to vCenter Server at that moment. When needed, you can perform a full database synchronization for the selected storage provider.
Remove	Unregister storage providers that you do not use. After this operation, vCenter Server closes the connection and removes the storage provider from its configuration.
Refresh certificate	vCenter Server warns you when a certificate assigned to a storage provider is about to expire. You can refresh the certificate to continue using the provider. If you fail to refresh the certificate before it expires, vCenter Server discontinues using the provider.
Upgrade Storage Provider	When vCenter Server notifies you about a new available VASA version, accept this version to enable it in your vSphere environment. vCenter Server activates new capabilities exposed by the storage provider and communicates this information to all hosts in the inventory. When you perform this action, you must enter credentials of the storage provider. You cannot select specific VASA version for an upgrade if several versions are available. vCenter Server upgrades only to the latest version. **Note** You have the flexibility to upgrade at your own pace. For more information, see vSphere Virtual Volumes Security.
Re-authenticate vCenter Server	Re-authenticate vCenter Server against the storage provider. This action re-authorizes vCenter Server with the storage provider if it has lost authentication and would need credentials. You must enter credentials of the storage provider.

Re-authenticate Host VASA Client

Starting from vSphere 8.0 Update 3, you can monitor the storage provider connection and authentication status on each ESXi host. If there is an authentication issue in a specific host, you can re-authenticate the specific host against the storage provider.

Use the vSphere Client to monitor the storage providers.

Prerequisites

Register storage providers. See Register Storage Providers for Virtual Volumes.

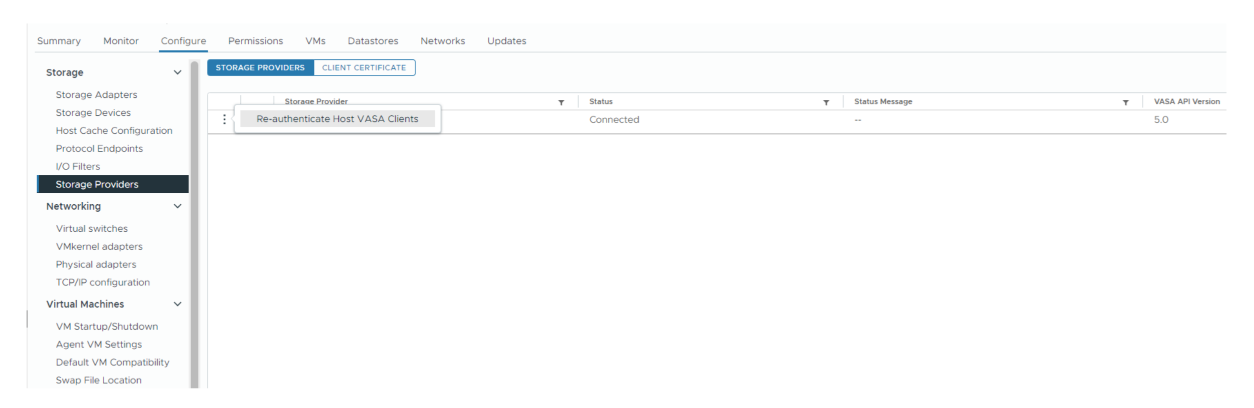
Procedure

1 Navigate to the ESXi host in vCenter Server.

2 Click the **Configure** tab, and click **Storage Providers**.

The **Storage Providers** tab lists the storage provider registered with the ESXi host. You can view general information including the name of the storage provider, status, and version of the VASA API. The **Client Certificate** lists the VASA client certificate details.

3 If the status indicates an authentication error for a provider, click the row action (three dots) icon next to the provider's name and click **Re-authenticate Host VASA Clients**.



The VASA client re-authorizes the ESXi host with the storage provider.

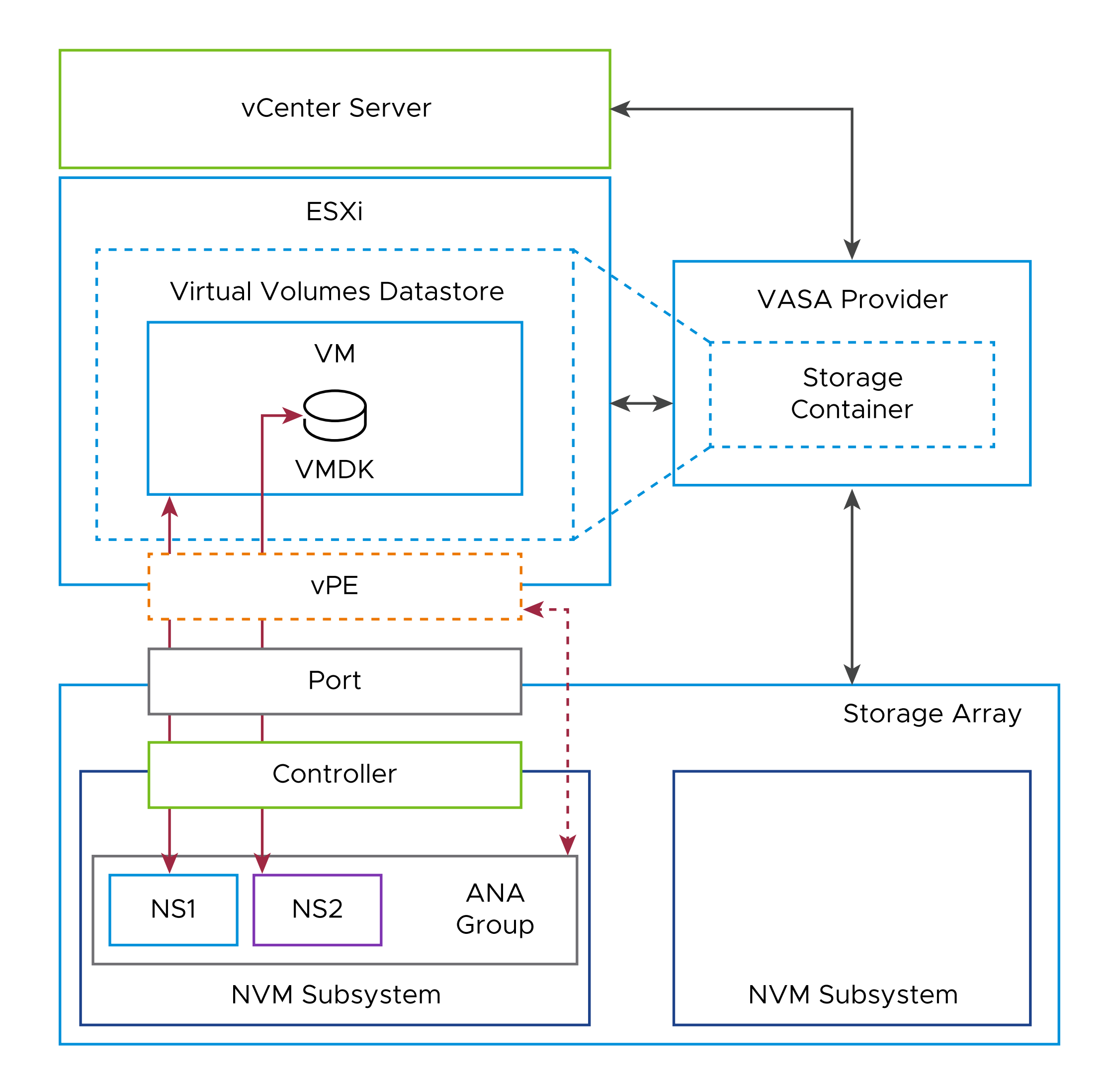
NVMe and Virtual Volumes in vSphere

In vSphere, Virtual Volumes supports NVMe over Fibre Channel and NVMe over TCP protocols.

When you use this protocol, after you register the storage provider (VASA), the system automatically completes the underlying setup. You only create the datastore. The virtual protocol endpoints and connections are handled by the storage provider.

Note Starting with vSphere 8.0 U2, third-party MPPs support NVMe with Virtual Volumes. For more information about MPPs, see Multipathing Concepts and Acronyms.

Be familiar with the following concepts specific to the configuration of NVMe with Virtual Volumes.



Virtual Volume Namespace

Generally, a namespace can be considered an NVMe equivalent of a LUN. Each type of virtual volume, such as data, configuration, or swap virtual volume, creates a namespace with a 1:1 virtual volume to namespace ratio. The namespace limits depend on the array vendor.

Asymmetrical Namespace Access (ANA) Group

With SCSI or NFS Virtual Volumes implementation, the storage container logically groups the virtual volume objects. With NVMe, grouping of the virtual volumes depends on the array. Generally, at the array, the virtual volumes namespaces are grouped into an ANA group. The array determines the number of ANA groups, each having a unique ANAGRPID within the NVM subsystem. Virtual volume namespaces are allocated and active only on BIND request to the VASA provider. The BIND request also adds the namespace to the ANA group. A namespace remains allocated and active until the last host unbinds the virtual volume.

NVMe Virtual Protocol Endpoint

With SCSI or NFS protocols, protocol endpoints represent a proxy LUN or an NFS mount point. These protocol points are called static.

Unlike a static protocol point, the NVMe protocol endpoint is virtual. This protocol endpoint is a logical representation of the ANA group where the virtual volumes reside. Until a VM is powered on, the NVMe protocol point doesn't exist. Once the VM is powered on, the NVMe protocol point is created, so that the host can access the virtual volumes in the ANA group.

For information about viewing the NVMe virtual protocol endpoints in the vSphere Client, see Review NVMe Protocol Endpoints.

Configure NVMe with Virtual Volumes

Follow these steps to configure NVMe with Virtual Volumes on your ESXi host.

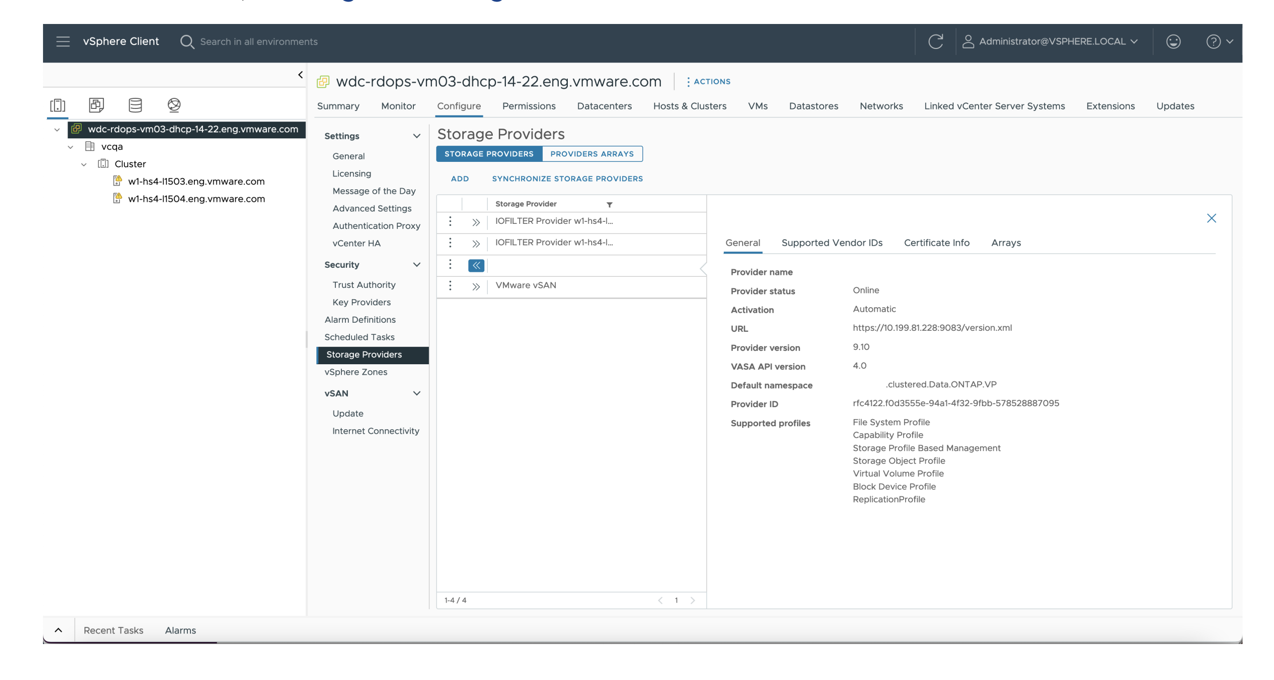
Prerequisites

- Make sure that your storage provider (VASA) supports NVMe with Virtual Volumes.
- Make sure that your storage array supports the NVMe with Virtual Volumes capability.
- Prepare your vSphere environment and a storage system. For more information, see Before You Enable vSphere Virtual Volumes and Requirements for VMware NVMe Storage.
- If you use NVMe over TCP, configure adapters for NVMe over TCP storage. See Configuring NVMe over TCP on ESXi.

Procedure

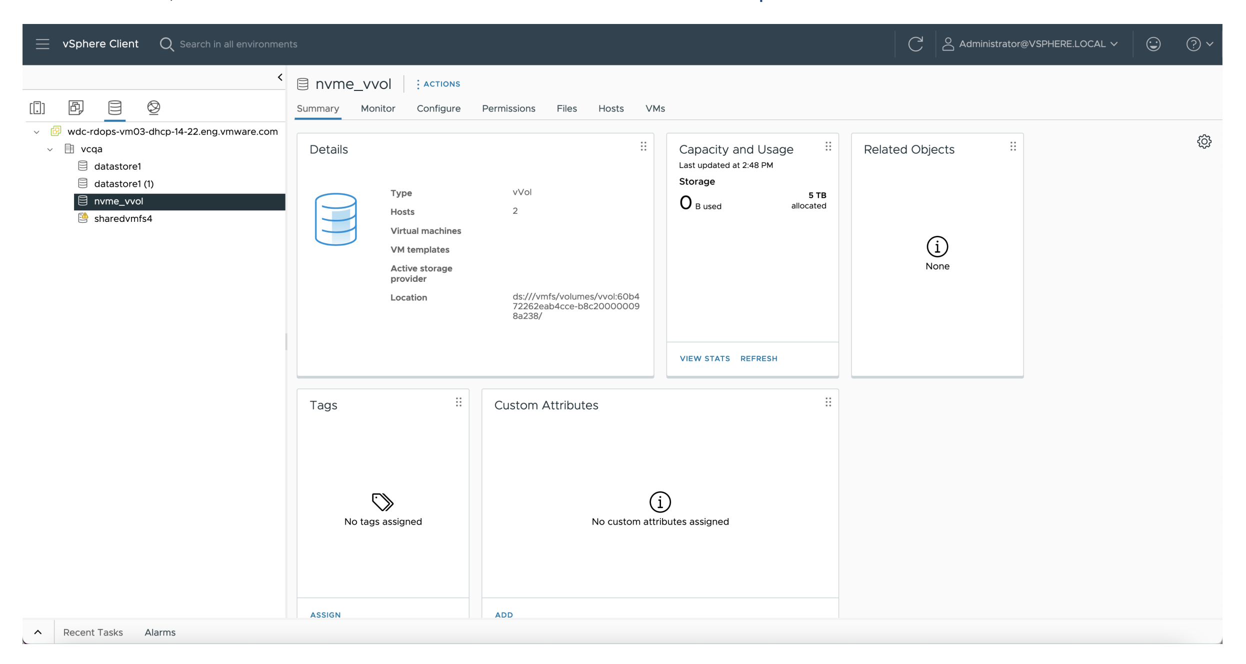
1 Register the storage provider.

For information, see Register Storage Providers for Virtual Volumes.



2 Create a Virtual Volumes datastore.

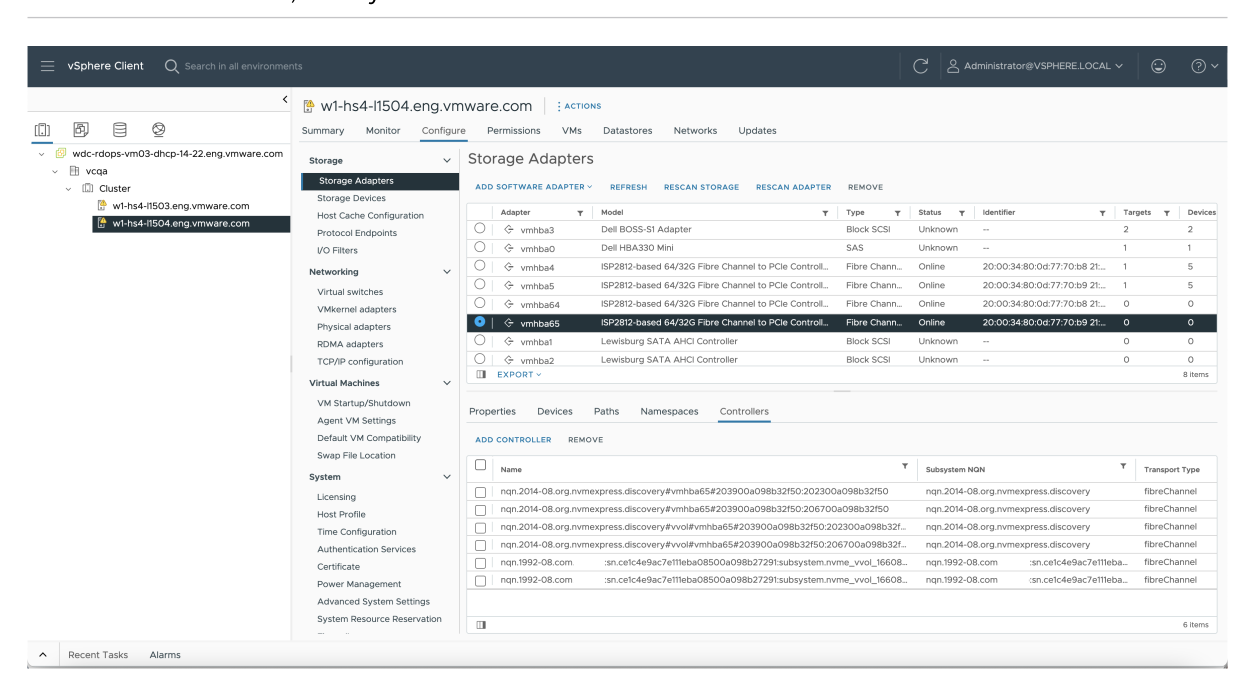
When creating the datastore, select the storage containers that support NVMe capability. For information, see Create a Virtual Volumes Datastore in vSphere Environment.



3 If you use NVMe over TCP, add a controller.

See Add Controllers for NVMe over Fabrics.

Note You do not need to configure the controller for NVMe over Fibre Channel. After you create the datastore, the system discovers controllers.



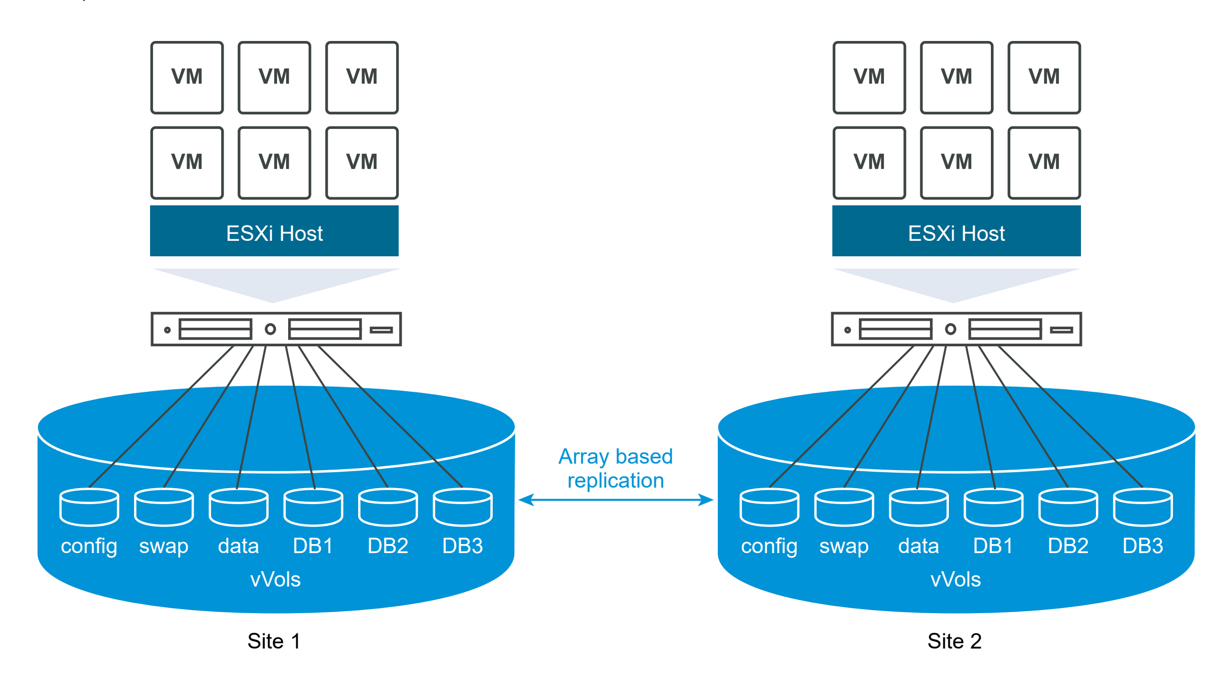
vSphere Virtual Volumes and Replication

vSphere Virtual Volumes support replication and disaster recovery. With the array-based replication, you can off-load replication of virtual machines to your storage array and use full replication capabilities of the array. You can replicate a single VM object, such as a virtual disk. You can also group several VM objects or virtual machines to replicate them as a single unit.

Array-based replication is policy driven. After you configure your Virtual Volumes storage for replication, information about replication capabilities and replication groups is delivered from the array by the storage provider. This information shows in the VM Storage Policy interface of vCenter Server.

You use the VM storage policy to describe replication requirements for your virtual machines. The parameters that you specify in the storage policy depend on how your array implements replication. For example, your VM storage policy might include such parameters as the replication schedule, replication frequency, or recovery point objective (RPO). The policy might also indicate the replication target, a secondary site where your virtual machines are replicated, or specify whether replicas must be deleted.

By assigning the replication policy during VM provisioning, you request replication services for your virtual machine. After that, the array takes over the management of all replication schedules and processes.



Requirements for Replication with Virtual Volumes

When you enable Virtual Volumes with replication, in addition to general Virtual Volumes requirements, your environment must satisfy several specific prerequisites.

For general Virtual Volumes requirements, see Before You Enable vSphere Virtual Volumes.

Storage Requirements

Implementation of Virtual Volumes replication depends on your array and might be different for storage vendors. Generally, the following requirements apply to all vendors.

- The storage arrays that you use to implement replication must be compatible with Virtual Volumes.
- The arrays must integrate with the version of the storage (VASA) provider compatible with Virtual Volumes replication.
- The storage arrays must be replication capable and configured to use vendor-provided replication mechanisms. Typical configurations usually involve one or two replication targets. Any required configurations, such as pairing of the replicated site and the target site, must be also performed on the storage side.
- When applicable, replication groups and fault domains for Virtual Volumes must be preconfigured on the storage side.

For more information, contact your vendor and see *VMware Compatibility Guide*.

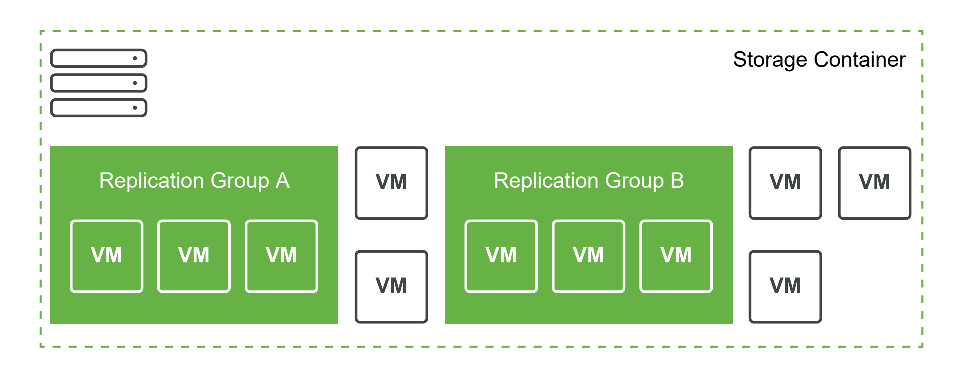
vSphere Requirements

- Use the vCenter Server and ESXi versions that support Virtual Volumes storage replication. vCenter Server and ESXi hosts that are older than 6.5 release do not support replicated Virtual Volumes storage. Any attempts to create a replicated VM on an incompatible host fail with an error. For information, see *VMware Compatibility Guide*.
- If you plan to migrate a virtual machine, make sure that target resources, such as the ESXi hosts and Virtual Volumes datastores, support storage replication.

Virtual Volumes and Replication Groups

When your storage offers replication services, in addition to storage containers and protocol endpoints, your storage administrator can configure replication groups on the storage side.

vCenter Server and ESXi can discover replication groups, but do not manage their life cycle. Replication groups, also called consistency groups, indicate which VMs and virtual disks must be replicated together to a target site. A single VM cannot span multiple replication groups.



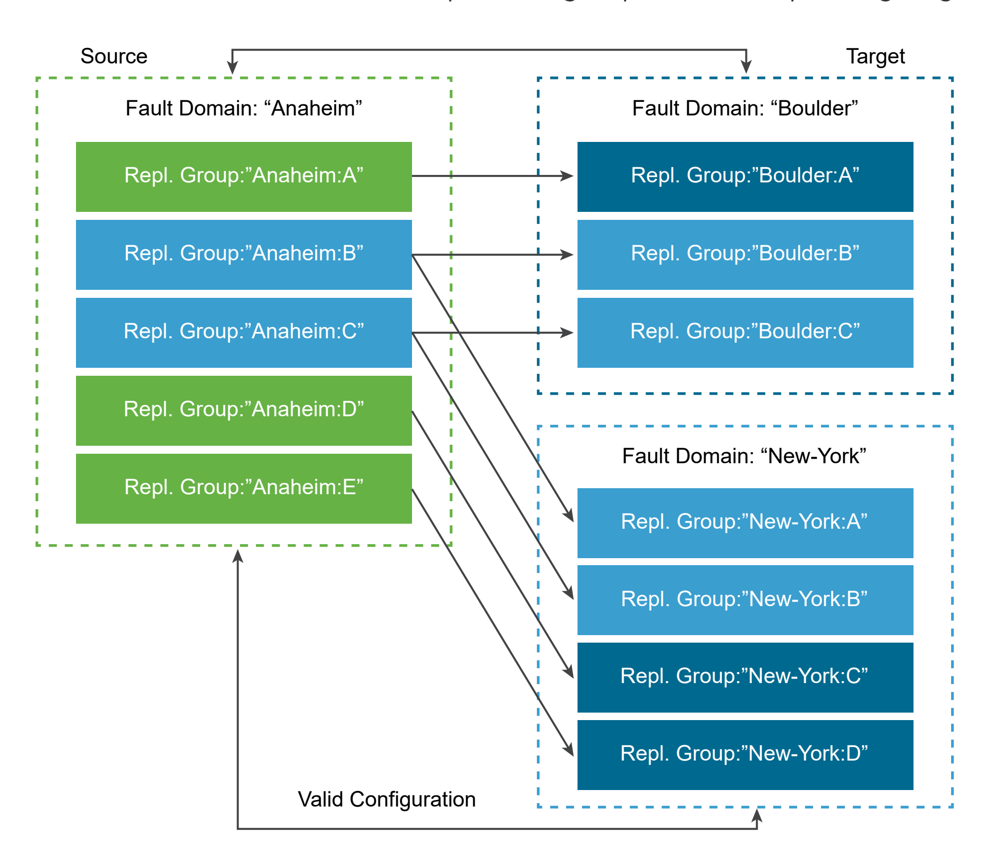
If no preconfigured groups are available, Virtual Volumes can use an automatic method. With the automatic method, Virtual Volumes creates a replication group on demand and associates this group with a Virtual Volumes object being provisioned. If you use the automatic replication group, all components of a virtual machine are assigned to the group. You cannot mix preconfigured and automatic replication groups for components of the same virtual machine.

Virtual Volumes and Fault Domains

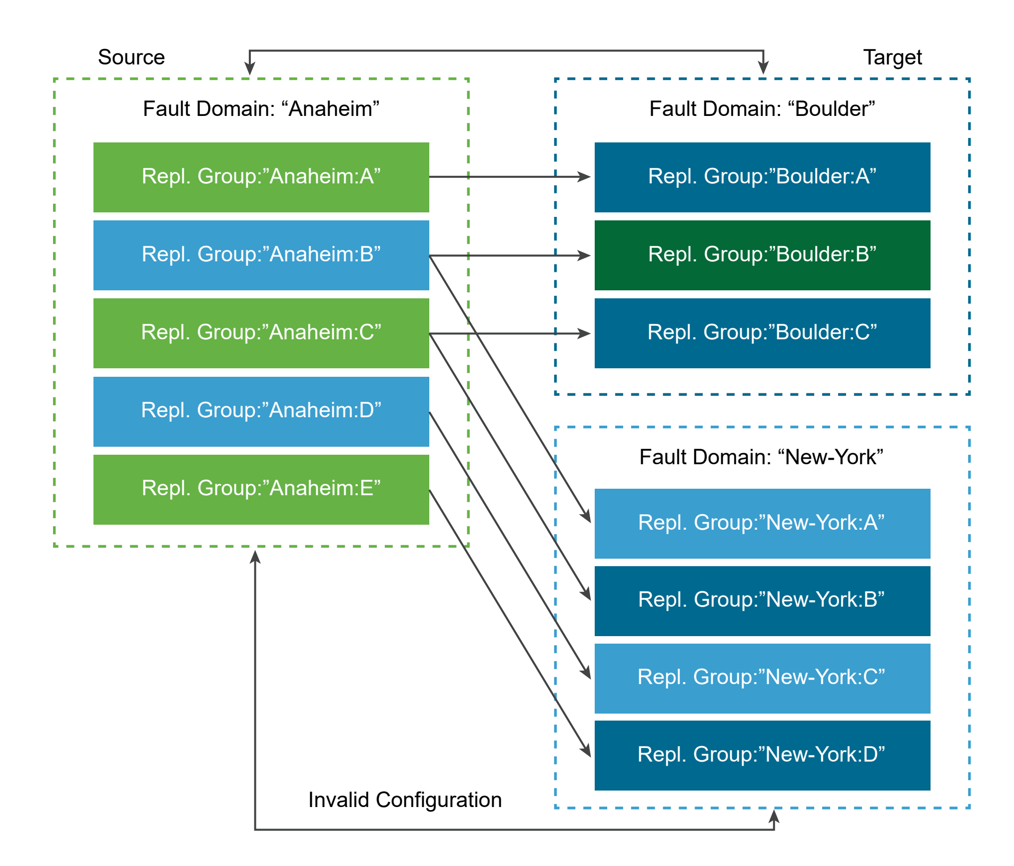
In the Virtual Volumes environment, fault domains define how specific replication groups must be combined when being replicated from a source to a target site.

Fault domains are configured and reported by the storage array, and are not exposed in the vSphere Client. The Storage Policy Based Management (SPBM) mechanism discovers fault domains and uses them for validation purposes during a virtual machine creation.

For example, provision two virtual machines, one associated with replication group Anaheim: B, the second associated with replication group Anaheim: C. SPBM validates the provisioning because each VM is in its own replication group with corresponding target fault domains.



Now provision two VMs, one associated with replication group Anaheim: B, the second associated with replication group Anaheim: D. This configuration is invalid. Both replication groups replicate to the New-York fault domain, however, only one replicates to the Boulder fault domain.



Virtual Volumes Replication Workflow

If information about replication capabilities of the Virtual Volumes storage array shows in vCenter Server, you can activate replication for your virtual machines.

The workflow to activate replication for your virtual machines includes steps typical for the virtual machine provisioning on Virtual Volumes storage.

1 Define the VM storage policy compatible with replication storage. The datastore-based rules of the policy must include the replication component. See Create a VM Storage Policy for Virtual Volumes.

  After you configure the storage policy that includes replication, vCenter Server discovers available replication groups.

2 Assign the replication policy to your virtual machine. If configured, select a compatible replication group, or use the automatic assignment. See Assign Storage Policies to Virtual Machines.

Replication Guidelines and Considerations

When you use replication with Virtual Volumes, specific considerations apply.

- You can apply the replication storage policy only to a configuration virtual volume and a data virtual volume. Other VM objects inherit the replication policy in the following way:
  - The memory virtual volume inherits the policy of the configuration virtual volume.

  - The digest virtual volume inherits the policy of the data virtual volume.
  - The swap virtual volume, which exists while a virtual machine is powered on, is excluded from replication.
- If you do not apply the replication policy to a VM disk, the disk is not replicated.
- The replication storage policy should not be used as a default storage policy for a datastore. Otherwise, the policy prevents you from selecting replication groups.
- Replication preserves snapshot history. If a snapshot was created and replicated, you can recover to the application consistent snapshot.
- You can replicate a linked clone. If a linked clone is replicated without its parent, it becomes a full clone.
- If a descriptor file belongs to a virtual disk of one VM, but resides in the VM home of another VM, both VMs must be in the same replication group. If the VMs are located in different replication groups, both of these replication groups must be failed over at the same time. Otherwise, the descriptor might become unavailable after the failover. As a result, the VM might fail to power on.
- In your Virtual Volumes with replication environment, you might periodically run a test failover workflow to ensure that the recovered workloads are functional after a failover.

  The resulting test VMs that are created during the test failover are fully functional and suitable for general administrative operations. However, certain considerations apply:

  - All VMs created during the test failover must be deleted before the test failover stops. The deletion ensures that any snapshots or snapshot-related virtual volumes that are part of the VM, such as the snapshot virtual volume, do not interfere with stopping of the test failover.
  - You can create full clones of the test VMs.
  - You can create fast clones only if the policy applied to the new VM contains the same replication group ID as the VM being cloned. Attempts to place the child VM outside of the replication group of the parent VM fail.

Best Practices for Working with vSphere Virtual Volumes

Observe the following recommendations when you use Virtual Volumes with ESXi and vCenter Server.

Guidelines and Limitations when Using Virtual Volumes

For the best experience with Virtual Volumes functionality, you must follow specific guidelines.

Virtual Volumes supports the following capabilities, features, and VMware products:

- With Virtual Volumes, you can use advanced storage services that include replication, encryption, deduplication, and compression on individual virtual disks. Contact your storage vendor for information about services they support with Virtual Volumes.
- Virtual Volumes functionality supports backup software that uses vSphere APIs - Data Protection. Virtual volumes are modeled on virtual disks. Backup products that use vSphere APIs - Data Protection are as fully supported on virtual volumes as they are on VMDK files on a LUN. Snapshots that the backup software creates using vSphere APIs - Data Protection look as non-vVols snapshots to vSphere and the backup software.

  Note Virtual Volumes do not support SAN transport mode. vSphere APIs - Data Protection automatically selects an alternative data transfer method.

  For more information about integration with the vSphere Storage APIs - Data Protection, consult your backup software vendor.
- Virtual Volumes supports vSphere features such as vSphere vMotion, Storage vMotion, snapshots, linked clones, and DRS.
- You can use clustering products, such as Oracle Real Application Clusters, with Virtual Volumes. To use these products, you activate the multiwrite setting for a virtual disk stored on the Virtual Volumes datastore.

For more details, see the knowledge base article at http://kb.vmware.com/kb/2112039. For a list of features and products that Virtual Volumes functionality supports, see *VMware Product Interoperability Matrixes*.

Virtual Volumes Limitations

Improve your experience with Virtual Volumes by knowing the following limitations:

- Because the Virtual Volumes environment requires vCenter Server, you cannot use Virtual Volumes with a standalone host.
- Virtual Volumes functionality does not support RDMs.
- A Virtual Volumes storage container cannot span multiple physical arrays. Some vendors present multiple physical arrays as a single array. In such cases, you still technically use one logical array.
- Host profiles that contain Virtual Volumes datastores are vCenter Server specific. After you extract this type of host profile, you can attach it only to hosts and clusters managed by the same vCenter Server as the reference host. For more information, see *Configuration Maximums* document.

Virtual Volumes Limitations with NVMe

The following features are not supported for NVMe over Virtual Volumes specific to vSphere 8.0 release:

- Replication and Site Recovery Manager operations.
- NVMe reservations.
- NVMe deallocate and unmap operations.
- Disk sharing across VMs.
- NVMe in-band migration.

Best Practices for Storage Container Provisioning

Follow these best practices when provisioning storage containers on the Virtual Volumes array side.

Creating Containers Based on Your Limits

Because storage containers apply logical limits when grouping virtual volumes, the container must match the boundaries that you want to apply.

Examples might include a container created for a tenant in a multitenant deployment, or a container for a department in an enterprise deployment.

- Organizations or departments, for example, Human Resources and Finance
- Groups or projects, for example, Team A and Red Team
- Customers

Putting All Storage Capabilities in a Single Container

Storage containers are individual datastores. A single storage container can export multiple storage capability profiles. As a result, virtual machines with diverse needs and different storage policy settings can be a part of the same storage container.

Changing storage profiles must be an array-side operation, not a storage migration to another container.

Avoiding Over-Provisioning Your Storage Containers

When you provision a storage container, the space limits that you apply as part of the container configuration are only logical limits. Do not provision the container larger than necessary for the anticipated use. If you later increase the size of the container, you do not need to reformat or repartition it.

Using Storage-Specific Management UI to Provision Protocol Endpoints

Every storage container needs protocol endpoints (PEs) that are accessible to ESXi hosts.

When you use block storage, the PE represents a proxy LUN defined by a T10-based LUN WWN. For NFS storage, the PE is a mount point, such as an IP address or DNS name, and a share name. For NVMe storage, the PE is an ANA group and proxy namespaces are virtual.

Typically, configuration of PEs is array-specific. When you configure PEs, you might need to associate them with specific storage processors, or with certain hosts. To avoid errors when creating PEs, do not configure them manually. Instead, when possible, use storage-specific management tools.

No Assignment of IDs Above Disk.MaxLUN to Protocol Endpoint LUNs

By default, an ESXi host can access LUN IDs that are within the range of 0 to 1023. If the ID of the protocol endpoint LUN that you configure is 1024 or greater, the host might ignore the PE.

If your environment uses LUN IDs that are greater than 1023, change the number of scanned LUNs through the `Disk.MaxLUN` parameter. See Change the Number of Scanned Storage Devices.

Best Practices for Virtual Volumes Performance

To ensure optimal Virtual Volumes performance results, follow these recommendations.

Using Different VM Storage Policies for Individual Virtual Volume Components

By default, all components of a virtual machine in the Virtual Volumes environment get a single VM storage policy. However, different components might have different performance characteristics, for example, a database virtual disk and a corresponding log virtual disk. Depending on performance requirements, you can assign different VM storage policies to individual virtual disks and to the VM home file, or config-vVol.

When you use the vSphere Client, you cannot change the VM storage policy assignment for swap-vVol, memory-vVol, or snapshot-vVol.

See Create a VM Storage Policy for Virtual Volumes.

Getting a Host Profile with Virtual Volumes

The best way to get a host profile with Virtual Volumes is to configure a reference host and extract its profile. If you manually edit an existing host profile in the vSphere Client and attach the edited profile to a new host, you might trigger compliance errors. Other unpredictable problems might occur. For more details, see the VMware Knowledge Base article 2146394.

Monitoring I/O Load on Individual Protocol Endpoint

- All virtual volume I/O goes through protocol endpoints (PEs). Arrays select protocol endpoints from several PEs that are accessible to an ESXi host. Arrays can do load balancing and change the binding path that connects the virtual volume and the PE. See Binding and Unbinding Virtual Volumes.

- On block storage, ESXi gives a large queue depth to I/O because of a potentially high number of virtual volumes. The `Scsi.ScsiVVolPESNRO` parameter controls the number of I/O that can be queued for PEs. You can configure the parameter on the Advanced System Settings page of the vSphere Client.

Monitoring Array Limitations

A single VM might occupy multiple virtual volumes. See Virtual Volume Objects.

Suppose that your VM has two virtual disks, and you take two snapshots with memory. Your VM might occupy up to 10 Virtual Volumes objects: a config-vVol, a swap-vVol, two data-vVols, four snapshot-vVols, and two memory snapshot-vVols.

Ensuring that Storage Provider Is Available

To access Virtual Volumes storage, your ESXi host requires a storage provider (VASA provider). It is either built into the storage array firmware or deployed as an appliance in the form of a VM. If your VASA provider is a VM based appliance, the storage provider is always available. To ensure that the storage provider is always available, follow these guidelines:

- Do not migrate a storage provider VM to Virtual Volumes storage.
- Back up your storage provider VM.
- When appropriate, use vSphere HA or Site Recovery Manager to protect the storage provider VM.

Using Stretched Storage Clustering with Virtual Volumes

Starting with vSphere 8.0 Update 3, Virtual Volumes support Stretched Storage Clustering (SSC) through SCSI.

SSC is a specific storage configuration that is commonly referred to as Metro Storage Clustering. A metro cluster is a host cluster that spans multiple sites located in different locations (usually two). The different sites that make up these types of clusters are almost always located in the same area if not city due to the latency requirements. The hosts may be in different data centers but still access the same shared storage. SSC configurations allow fully active and workload-balanced data centers to be used to their full potential and for fast recovery in the event of a host or even full site failure. These configurations are usually implemented in environments where disaster and downtime avoidance is a key requirement.

Virtual Volumes support the uniform host access SSC configuration where vSphere hosts from both sites are all connected to a storage node in the storage cluster across all sites. For information about specific requirements and best practices for an SSC implementation, see https://core.vmware.com/vsphere-virtual-volumes-vvols.

Using the `esxcli storage vvol` Command to Troubleshoot vSphere Virtual Volumes

Learn about the command options that the `esxcli storage vvol` takes. The topics also provide solutions to problems that you might encounter when using vSphere Virtual Volumes.

Virtual Volumes and esxcli Commands

You can use the `esxcli storage vvol` commands to troubleshoot your Virtual Volumes environment.

The following command options are available.

Table 14-1. `esxcli storage vvol` commands

Namespace	Command Option	Description
`esxcli storage core device`	list	Identify protocol endpoints. The output entry `Is VVOL PE: true` indicates that the storage device is a protocol endpoint.
`esxcli storage vvol nvme info`	get	Get information for the Virtual Volumes NVMe host information.
`esxcli storage vvol storagearray`	list	List containers or protocol endpoints on the storage array.
`esxcli storage vvol storagecontainer`	list	List all available storage containers.
	abandonedvvol scan	Scan the specified storage container for abandoned Virtual Volumes.
`esxcli storage vvol daemon`	unbindall	Unbind all virtual volumes from all VASA providers known to the ESXi host.
`esxcli storage vvol protocolendpoint`	list	List all protocol endpoints that your host can access.
	startforceunbind	Unbind all Virtual Volumes on the PE with zero open handle count.
`esxcli storage vvol stats`		Track performance statistics. For information, see Collecting Statistical Information for Virtual Volumes.
`esxcli storage vvol vasacontext`	get	Show the VASA context (VC UUID) associated with the host.
`esxcli storage vvol vasaprovider`	list	List all storage (VASA) providers associated with the host.

Table 14-1. esxcli storage vvol commands (continued)

Namespace	Command Option	Description
esxcli storage vvol virtualvolume	unmap	Reclaim the space by unmapping free blocks from VMFS config volumes. For information, see Reclaim Space on the vSphere Virtual Volumes Datastores.
esxcli storage vvol vmstats	get	Get the Virtual Volumes information and statistics for a specific virtual machine.

Collecting Statistical Information for Virtual Volumes

You can use the vvol stats command in your ESXi host to track performance statistics.

The following command options are available.

Command	Description	Options
esxcli storage vvol stats get	Get statistics for all VASA providers (default), or for specified namespace or entity in the given namespace.	-e|--entity=*str* Enter entity ID. -n|--namespace=*str* Enter node namespace expression. -r|--raw Use raw format output.
esxcli storage vvol stats list	List all the statistics nodes (default), or nodes under a specified namespace.	-n|--namespace=*str* Enter node namespace expression.
esxcli storage vvol stats enable	Activate statistics tracking for the complete namespace.	
esxcli storage vvol stats disable	Deactivate statistics tracking for the complete namespace.	
esxcli storage vvol stats add	Activate statistics tracking for a specific entity under a specific namespace.	-e|--entity=*str* Enter entity ID. -n|--namespace=*str* Enter node namespace expression.
esxcli storage vvol stats remove	Removes specific entity for statistics tracking under the specified namespace.	-e|--entity=*str* Enter entity ID. -n|--namespace=*str* Enter node namespace expression.
esxcli storage vvol stats reset	Reset the statistics counter for the specified statistics namespace or entity.	-e|--entity=*str* Enter entity ID. -n|--namespace=*str* Enter node namespace expression.

Virtual Volumes Datastore Is Inaccessible

After you create a Virtual Volumes datastore, it remains inaccessible.

Problem

The vSphere Client shows the datastore as inaccessible. You cannot use the datastore for virtual machine provisioning.

Cause

This problem might occur when you fail to configure protocol endpoints for the SCSI-based storage container or NVMe ANA groups mapped to the virtual datastore. Like traditional LUNs, SCSI protocol endpoints need to be configured so that an ESXi host can detect them.

Solution

Before creating virtual datastores for SCSI-based containers, make sure to configure protocol endpoints on the storage side.

Failures When Migrating VMs or Deploying VM OVFs to Virtual Volumes Datastores

Your attempts to migrate a virtual machine or to deploy a VM OVF to Virtual Volumes datastores fail.

Problem

An OVF template or a VM being migrated from a nonvirtual datastore might include additional large files, such as ISO disk images, DVD images, and image files. If these additional files cause the configuration virtual volume to exceed its 4-GB limit, migration or deployment to a virtual datastore fails.

Cause

The configuration virtual volume, or config-vVol, contains various VM-related files. On traditional nonvirtual datastores, these files are stored in the VM home directory. Similar to the VM home directory, the config-vVol typically includes the VM configuration file, virtual disk and snapshot descriptor files, log files, lock files, and so on.

On virtual datastores, all other large-sized files, such as virtual disks, memory snapshots, swap, and digest, are stored as separate virtual volumes.

Config-vVols are created as 4-GB virtual volumes. Generic content of the config-vVol usually consumes only a fraction of this 4-GB allocation, so config-vVols are typically thin-provisioned to conserve backing space. Any additional large files, such as ISO disk images, DVD images, and image files, might cause the config-vVol to exceed its 4-GB limit. If such files are included in an OVF template, deployment of the VM OVF to Virtual Volumes storage fails. If these files are part of an existing VM, migration of that VM from a traditional datastore to Virtual Volumes storage also fails.

Note Starting with vSphere 7.0 Update 2, partners can increase the config-vVol to above 4 GB. Work with your Virtual Volumes partner on implementing this if it is supported by your partner and applicable to your environment.

Solution

- For VM migration. Before migrating a VM from a traditional datastore to a virtual datastore, remove excess content from the VM home directory to keep the config-vVol under the 4-GB limit.
- For OVF deployment. Because you cannot deploy an OVF template that contains excess files directly to a virtual datastore, first deploy the VM to a nonvirtual datastore. Remove any excess content from the VM home directory, and migrate the resulting VM to Virtual Volumes storage.

Filtering Virtual Machine I/O in vSphere

15

I/O filters are software components that can be installed on ESXi hosts and can offer additional data services to virtual machines. The filters process I/O requests, which move between the guest operating system of a virtual machine and virtual disks.

The I/O filters can be offered by VMware or created by third parties through vSphere APIs for I/O Filtering (VAIO).

About I/O Filters

I/O filters can gain direct access to the virtual machine I/O path. You can enable the I/O filter for an individual virtual disk. The I/O filters are independent of the storage topology.

VMware offers certain categories of I/O filters. In addition, third-party vendors can create the I/O filters. Typically, they are distributed as packages that provide an installer to deploy the filter components on vCenter Server and ESXi host clusters.

After the I/O filters are deployed, vCenter Server configures and registers an I/O filter storage provider, also called a VASA provider, for each host in the cluster. The storage providers communicate with vCenter Server and make data services offered by the I/O filter visible in the VM Storage Policies interface. You can reference these data services when defining common rules for a VM policy. After you associate virtual disks with this policy, the I/O filters are enabled on the virtual disks.

Datastore Support

I/O filters can support all datastore types including the following:

- VMFS
- NFS 3
- NFS 4.1
- vVol
- vSAN

System Requirements for I/O Filters

To be able to use I/O filters in your environment, you must follow specific requirements.

- Use the latest version of ESXi and vCenter Server compatible with I/O filters. Older versions might not support I/O filters, or provide only partial support.
- Check for any additional requirements that individual partner solutions might have. In specific cases, your environment might need flash devices, extra physical memory, or network connectivity and bandwidth. For information, contact your vendor or your VMware representative.
- Web server to host partner packages for filter installation. The server must remain available after initial installation. When a new host joins the cluster, the server pushes appropriate I/O filter components to the host.

I/O Filter Guidelines and Best Practices

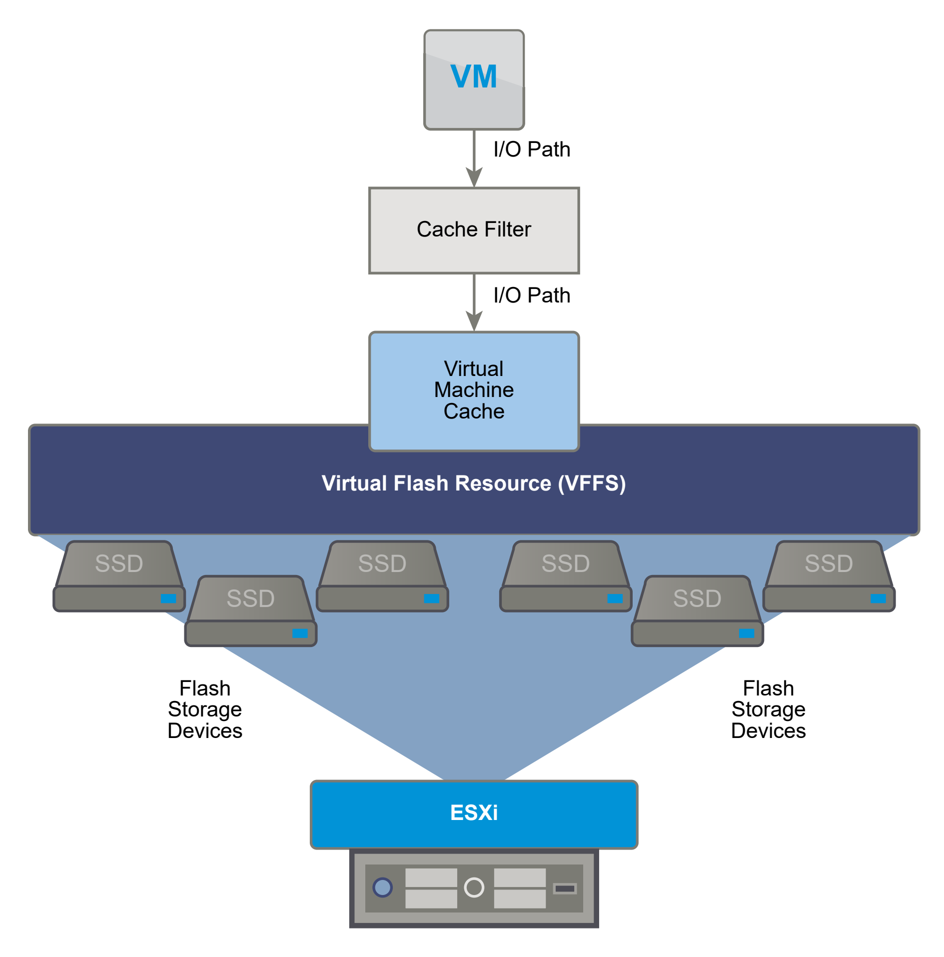
When you use I/O filters in your environment, follow specific guidelines and best practices.

- Because I/O filters are datastore-agnostic, all types of datastores, including VMFS, NFS, Virtual Volumes, and vSAN, are compatible with I/O filters.
- I/O filters support RDMs in virtual compatibility mode. No support is provided to RDMs in physical compatibility mode.
- You cannot change or assign the I/O filter policy while migrating or cloning a virtual machine. You can change the policy after you complete the migration or cloning.
- When you clone or migrate a virtual machine with I/O filter policy from one host to another, make sure that the destination host has a compatible filter installed. This requirement applies to migrations initiated by an administrator or by such functionalities as HA or DRS.
- When you convert a template to a virtual machine, and the template is configured with I/O filter policy, the destination host must have the compatible I/O filter installed.
- If you use vCenter Site Recovery Manager to replicate virtual disks, the resulting disks on the recovery site do not have the I/O filter policies. You must create the I/O filter policies in the recovery site and reattach them to the replicated disks.
- If your virtual machine has a snapshot tree associated with it, you cannot add, change, or remove the I/O filter policy for the virtual machine.

Using Cache I/O Filters with Flash Storage Devices on ESXi Hosts

A cache I/O filter can use a local flash device on your ESXi host to cache virtual machine data.

If your caching I/O filter uses local flash devices, you need to configure a virtual flash resource, also known as VFFS volume. You configure the resource on your ESXi host before activating the filter. While processing the virtual machine read I/Os, the filter creates a virtual machine cache and places it on the VFFS volume.



To set up a virtual flash resource, you use flash devices that are connected to your host. To increase the capacity of your virtual flash resource, you can add more flash drives. An individual flash drive must be exclusively allocated to a virtual flash resource and cannot be shared with any other vSphere service, such as vSAN or VMFS. See Set Up Virtual Flash Resource.

Read the following topics next:

- Deploying and Managing I/O Filters in vSphere
- Activate I/O Filter Data Services on Virtual Disks in vSphere

Deploying and Managing I/O Filters in vSphere

You can run the installer provided by your vendor to install, uninstall, or upgrade I/O filters in your vSphere environment.

When you work with I/O filters, the following considerations apply:

- vCenter Server uses ESX Agent Manager (EAM) to install and uninstall I/O filters. As an administrator, never invoke EAM APIs directly for EAM agencies that are created or used by vCenter Server. All operations related to I/O filters must go through VIM APIs. If you accidentally modify an EAM agency that was created by vCenter Server, you must revert the changes. If you accidentally destroy an EAM agency that is used by I/O filters, you must call `Vim.IoFilterManager#uninstallIoFilter` to uninstall the affected I/O filters. After uninstalling, perform a fresh reinstall.
- When a new host joins the cluster that has I/O filters, the filters installed on the cluster are deployed on the host. vCenter Server registers the I/O filter storage provider for the host. Any cluster changes become visible in the VM Storage Policies interface of the vSphere Client.
- When you move a host out of a cluster or remove it from vCenter Server, the I/O filters are uninstalled from the host. vCenter Server unregisters the I/O filter storage provider.
- If you use a stateless ESXi host, it might lose its I/O filter VIBs during a reboot. vCenter Server checks the bundles installed on the host after it reboots, and pushes the I/O filter VIBs to the host if necessary.

Configure I/O Filters in the vSphere Environment

To set up data services that the I/O filters provide for your virtual machines, follow several steps.

Prerequisites

- Create a cluster that includes at least one ESXi host.
- For information about I/O filters provided by third parties, contact your vendor or your VMware representative.

Install I/O Filters in a Cluster

If you use I/O filters provided by third parties, install the I/O filters in an ESXi host cluster.

VMware partners create I/O filters through the vSphere APIs for I/O Filtering (VAIO) developer program.

The filter packages are distributed as solution bundle ZIP packages that can include I/O filter daemons, I/O filter libraries, CIM providers, and other associated components.

Typically, to deploy the filters, you run installers provided by vendors. Installation is performed at an ESXi cluster level. You cannot install the filters on selected hosts directly.

Prerequisites

- Required privileges: **Host.Configuration.Query patch**.
- Verify that the I/O filter solution is certified by VMware.

Procedure

- Run the installer that the vendor provided.

  The installer deploys the appropriate I/O filter extension on vCenter Server and the filter components on all hosts within a cluster.

  A storage provider, also called a VASA provider, is automatically registered for every ESXi host in the cluster. Successful auto-registration of the I/O filter storage providers triggers an event at the host level. If the storage providers fail to auto-register, the system raises alarms on the hosts.

View I/O Filters and Storage Providers

Use the vSphere Client to review I/O filters available in your environment and verify that the I/O filter providers appear as expected and are active.

When you install a third-party I/O filter, a storage provider, also called VASA provider, is automatically registered for every ESXi host in the cluster. Successful auto-registration of the I/O filter storage providers triggers an event at the host level. If the storage providers fail to auto-register, the system raises alarms on the hosts.

Procedure

1. Verify that the I/O filter storage providers appear as expected and are active.
   a. Navigate to vCenter Server.
   b. Click the **Configure** tab, and click **Storage Providers**.
   c. Review the storage providers for I/O filters.

   When the I/O filter providers are properly registered, capabilities and data services that the filters offer populate the VM Storage Policies interface.
2. Verify that the I/O filter components are listed on your cluster and ESXi hosts.

Option	Actions
View I/O filters on a cluster	a Navigate to the cluster. b Click the **Configure** tab. c Under **Configuration**, click **I/O Filters** to review the filters installed in the cluster.
View I/O filters on a host	a Navigate to the host. b Click the **Configure** tab. c Under **Storage**, click **I/O Filters** to review the filters installed on the host.

Handling I/O Filter Installation Failures

Typically, all ESXi hosts in a cluster have the same set of I/O filters installed. Occasionally, failures might happen during installation.

If an I/O filter installation fails on a host, the system generates events that report the failure. In addition, an alarm on the host shows the reason for the failure. Examples of failures include the following:

- The VIB URL is not accessible from the host.
- The VIB has an invalid format.
- The VIB requires the host to be in maintenance mode for an upgrade or uninstallation.
- The VIB requires the host to reboot after the installation or uninstallation.
- Attempts to put the host in maintenance mode fail because the virtual machine cannot be evacuated from the host.
- The VIB requires manual installation or uninstallation.

vCenter Server can resolve some failures. You might have to intervene for other failures. For example, you might need to edit the VIB URL, manually evacuate or power off virtual machines, or manually install or uninstall VIBs.

Install I/O Filters on a Single ESXi Host

For troubleshooting purposes, you can download an ESXi component of the I/O filter, packaged as a VIB file, and install it on the ESXi host. Use the `esxcli` command to install the VIB file.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Install the VIBs by running the following command:

   `esxcli software vib install --depot` ***`path_to_VMware_vib_ZIP_file`***

   Options for the `install` command allow you to perform a dry run, specify a specific VIB, bypass acceptance-level verification, and so on. Do not bypass verification on production systems. See the *ESXCLI Reference* documentation.

2 Verify that the VIBs are installed on your ESXi host.

   `esxcli software vib list`

Uninstall I/O Filters from a Cluster

You can uninstall I/O filters deployed in an ESXi host cluster.

Prerequisites

- Required privileges: **Host.Config.Patch**.

Procedure

1 Uninstall the I/O filter by running the installer that your vendor provides.

   During uninstallation, a third party I/O filter installer automatically places the hosts into maintenance mode.

   If the uninstallation is successful, the filter and any related components are removed from the hosts.

2 Verify that the I/O filter components are properly uninstalled from your ESXi hosts. Use one of the following methods:

   - Run the `esxcli software vib list` command.
   - View the I/O filters in the vSphere Client. See View I/O Filters and Storage Providers.

   The uninstalled filter no longer appears on the list.

Upgrade I/O Filters in a Cluster

After you upgrade your ESXi hosts, use installers provided by I/O filter vendors to upgrade I/O filters deployed in the ESXi host cluster.

An upgrade consists of uninstalling the old filter components and replacing them with the new filter components. To determine whether an installation is an upgrade, vCenter Server checks the names and versions of existing filters. If the existing filter names match the names of the new filters but have different versions, the installation is considered an upgrade.

Prerequisites

- Required privileges:**Host.Config.Patch**.
- Upgrade your hosts to ESXi 8.0. If you use vSphere Lifecycle Manager for the upgrade, see the *Managing Host and Cluster Lifecycle* documentation.

Procedure

1 To upgrade the filter, run the vendor-provided installer.

   During the upgrade, a third party I/O filter installer automatically places the hosts into maintenance mode.

   The installer identifies any existing filter components and removes them before installing the new filter components.

2 Verify that the I/O filter components are properly upgraded in your ESXi hosts. Use one of the following methods:

   - Run the `esxcli software vib list` command.
   - View the I/O filters in the vSphere Client. See View I/O Filters and Storage Providers.

Results

After the upgrade, the system places the hosts back into operational mode.

Activate I/O Filter Data Services on Virtual Disks in vSphere

In vSphere, activating data services that I/O filters provide is a two-step process. You create a virtual machine policy based on data services that the I/O filters provide, and then attach this policy to a virtual machine.

Prerequisites

For the cache I/O filters, configure the virtual flash resource on your ESXi host before activating the filter. See Set Up Virtual Flash Resource.

For information about cache I/O filters, see Using Cache I/O Filters with Flash Storage Devices on ESXi Hosts.

Procedure

1 Define a VM policy based on I/O filter services.

   Make sure that the virtual machine policy lists data services provided by the I/O filters.

   See Create a VM Storage Policy for Host-Based Data Services.

2 Assign the I/O filter policy to a virtual machine.

   To activate data services that the I/O filter provides, associate the I/O filter policy with virtual disks. You can assign the policy when you provision the virtual machine.

   See Assign the I/O Filter Policy to Virtual Machines.

Assign the I/O Filter Policy to Virtual Machines

To activate data services that I/O filters provide, associate the I/O filter policy with virtual disks. You can assign the policy when you create or edit a virtual machine.

You can assign the I/O filter policy during an initial deployment of a virtual machine. This topic describes how to assign the policy when you create a new virtual machine. For information about other deployment methods, see the *vSphere Virtual Machine Administration* documentation.

Note You cannot change or assign the I/O filter policy when migrating or cloning a virtual machine.

Prerequisites

Verify that the I/O filter is installed on the ESXi host where the virtual machine runs.

Procedure

1 Start the virtual machine provisioning process and follow the appropriate steps.

2 Assign the same storage policy to all virtual machine files and disks.

   a On the **Select storage** page, select a storage policy from the **VM Storage Policy** drop-down menu.

   b Select the datastore from the list of compatible datastores and click **Next**.

      The datastore becomes the destination storage resource for the virtual machine configuration file and all virtual disks. The policy also activates I/O filter services for the virtual disks.

3 Change the VM storage policy for the virtual disk.

   Use this option to enable I/O filters just for your virtual disks.

   a On the **Customize hardware** page, expand the **New hard disk** pane.

   b From the **VM storage policy** drop-down menu, select the storage policy to assign to the virtual disk.

   c (Optional) Change the storage location of the virtual disk.

      Use this option to store the virtual disk on a datastore other than the datastore where the VM configuration file resides.

4 Complete the virtual machine provisioning process.

Results

After you create the virtual machine, the **Summary** tab displays the assigned storage policies and their compliance status.

What to do next

You can later change the virtual policy assignment. See Change Storage Policy Assignment for Virtual Machine Files and Disks.

Migrating Virtual Machines with I/O Filters

When you migrate a virtual machine with I/O filters, specific considerations apply.

If you use Storage vMotion to migrate a virtual machine with I/O filters, a destination datastore must be connected to hosts with compatible I/O filters installed.

You might need to migrate a virtual machine with I/O filters across different types of datastores, for example between VMFS and Virtual Volumes. If you do so, make sure that the VM storage policy includes rule sets for every type of datastore you are planning to use. For example, if you migrate your virtual machine between the VMFS and Virtual Volumes datastores, create a mixed VM storage policy that includes the following rules:

- Common Rules for the I/O filters
- Rule Set 1 for the VMFS datastore. Because Storage Policy Based Management does not offer an explicit VMFS policy, the rule set must include tag-based rules for the VMFS datastore.
- Rule Set 2 for the Virtual Volumes datastore

When Storage vMotion migrates the virtual machine, the correct rule set that corresponds to the target datastore is selected. The I/O filter rules remain unchanged.

If you do not specify rules for datastores and define only Common Rules for the I/O filters, the system applies default storage policies for the datastores.

Storage Hardware Acceleration in vSphere

16

The hardware acceleration functionality enables the ESXi host to integrate with compliant storage systems. The host can offload certain virtual machine and storage management operations to the storage systems. With the storage hardware assistance, your host performs these operations faster and consumes less CPU, memory, and storage fabric bandwidth.

Block storage devices, Fibre Channel and iSCSI, and NAS devices support the hardware acceleration.

For additional details, see the VMware knowledge base article at http://kb.vmware.com/kb/1021976.

Hardware Acceleration Benefits

When the hardware acceleration functionality is supported, the host can get hardware assistance and perform the following tasks faster and more efficiently.

- Migrating virtual machines with Storage vMotion
- Deploying virtual machines from templates
- Cloning virtual machines or templates
- VMFS clustered locking and metadata operations for virtual machine files
- Provisioning thick virtual disks
- Creating fault-tolerant virtual machines
- Creating and cloning thick disks on NFS datastores

Hardware Acceleration Requirements

The hardware acceleration functionality works only if you use an appropriate host and storage array combination.

Table 16-1. Hardware Acceleration Storage Requirements

ESXi	Block Storage Devices	NAS Devices
ESXi	Support T10 SCSI standard, or block storage plug-ins for array integration (VAAI)	Support NAS plug-ins for array integration

Note If your SAN or NAS storage fabric uses an intermediate appliance in front of a storage system that supports hardware acceleration, the intermediate appliance must also support hardware acceleration and be properly certified. The intermediate appliance might be a storage virtualization appliance, I/O acceleration appliance, encryption appliance, and so on.

Hardware Acceleration Support Status

For each storage device and datastore, the vSphere Client display the hardware acceleration support status.

The status values are Unknown, Supported, and Not Supported. The initial value is Unknown.

For block devices, the status changes to Supported after the host successfully performs the offload operation. If the offload operation fails, the status changes to Not Supported. The status remains Unknown if the device provides partial hardware acceleration support.

With NAS, the status becomes Supported when the storage can perform at least one hardware offload operation.

When storage devices do not support or provide partial support for the host operations, your host reverts to its native methods to perform unsupported operations.

Hardware Acceleration Considerations

When you use the hardware acceleration functionality with ESXi, certain considerations apply.

Several reasons might cause a hardware-accelerated operation to fail.

For any primitive that the array does not implement, the array returns an error. The error triggers the ESXi host to attempt the operation using its native methods.

The VMFS data mover does not leverage hardware offloads and instead uses software data movement when one of the following occurs:

- The source and destination VMFS datastores have different block sizes.
- The source file type is RDM and the destination file type is non-RDM (regular file).
- The source VMDK type is eagerzeroedthick and the destination VMDK type is thin.
- The source or destination VMDK is in sparse or hosted format.
- The source virtual machine has a snapshot.

- The logical address and transfer length in the requested operation are not aligned to the minimum alignment required by the storage device. All datastores created with the vSphere Client are aligned automatically.
- The VMFS has multiple LUNs or extents, and they are on different arrays.

Hardware cloning between arrays, even within the same VMFS datastore, does not work.

Read the following topics next:

- vSphere Hardware Acceleration for Block Storage Devices
- vSphere Hardware Acceleration on NAS Devices

vSphere Hardware Acceleration for Block Storage Devices

With hardware acceleration, your ESXi host can integrate with block storage devices, Fibre Channel, or iSCSI, and use storage array operations that include full copy, block zeroing, and hardware assisted locking.

ESXi hardware acceleration supports the following array operations:

- Full copy, also called clone blocks or copy offload. Storage arrays can make full copies of data within the array without having the host read and write the data. This operation reduces the time and network load when cloning virtual machines, provisioning from a template, or migrating with vMotion.
- Block zeroing, also called write same. Storage arrays can zero out a large number of blocks to provide newly allocated storage, free of previously written data. This operation reduces the time and network load when creating virtual machines and formatting virtual disks.
- Hardware assisted locking, also called atomic test and set (ATS). Supports discrete virtual machine locking without use of SCSI reservations. This operation allows disk locking per sector, instead of the entire LUN as with SCSI reservations.

Check with your vendor for the hardware acceleration support. Certain storage arrays require that you activate the support on the storage side.

On your host, the hardware acceleration is available by default. If your storage does not support the hardware acceleration, you can deactivate it.

In addition to hardware acceleration support, ESXi includes support for array thin provisioning. For information, see ESXi and Array Thin Provisioning.

Deactivate Hardware Acceleration for Block Storage Devices

On your host, the hardware acceleration for block storage devices is activated by default. You can use the vSphere Client advanced settings to deactivate the hardware acceleration operations.

As with any advanced settings, before you deactivate the hardware acceleration, consult with the VMware support team.

Procedure

1 In the vSphere Client, navigate to the ESXi host.

2 Click the **Configure** tab.

3 Under **System**, click **Advanced System Settings**.

4 To deactivate the hardware acceleration, change the value for these options to 0:

   - VMFS3.HardwareAcceleratedLocking
   - DataMover.HardwareAcceleratedMove
   - DataMover.HardwareAcceleratedInit

Managing Hardware Acceleration on Block Storage Devices

To integrate with the block storage arrays, vSphere uses the ESXi extensions called Storage APIs - Array Integration (VAAI). With this integration, vSphere can use the array hardware operations.

In the vSphere 5.x and later releases, these extensions are implemented as the T10 SCSI commands. As a result, with the devices that support the T10 SCSI standard, your ESXi host can communicate directly and does not require the VAAI plug-ins.

If the device does not support T10 SCSI or provides partial support, ESXi reverts to using the VAAI plug-ins, installed on your host. The host can also use a combination of the T10 SCSI commands and plug-ins. The VAAI plug-ins are vendor-specific and can be either VMware or partner developed. To manage the VAAI capable device, your host attaches the VAAI filter and vendor-specific VAAI plug-in to the device.

For information about whether your storage requires VAAI plug-ins or supports hardware acceleration through T10 SCSI commands, see the *VMware Compatibility Guide* or contact your storage vendor.

You can use several `esxcli` commands to query storage devices for the hardware acceleration support information. For the devices that require the VAAI plug-ins, the claim rule commands are also available. For information about `esxcli` commands, see *Getting Started with ESXCLI*.

Display Hardware Acceleration Plug-Ins and Filter

To communicate with the devices that do not support the T10 SCSI standard, your host uses a single VAAI filter and a vendor-specific VAAI plug-in. Use the `esxcli` command to view the hardware acceleration filter and plug-ins currently loaded into your system.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- Run the **esxcli storage core plugin list --plugin-class=*value*** command.

  For *value*, enter one of the following parameters:

  - Type `VAAI` to display plug-ins.

    The output of this command is similar to the following example:

    ```
    #esxcli storage core plugin list --plugin-class=VAAI
    Plugin name         Plugin class
    VMW_VAAIP_EQL       VAAI
    VMW_VAAIP_NETAPP    VAAI
    VMW_VAAIP_CX        VAAI
    ```

  - Type `Filter` to display the Filter.

    The output of this command is similar to the following example:

    ```
    esxcli storage core plugin list --plugin-class=Filter
    Plugin name  Plugin class
    VAAI_FILTER  Filter
    ```

Verify Hardware Acceleration Support Status

Use the `esxcli` command to verify the hardware acceleration support status of a particular storage device.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- Run the **esxcli storage core device list -d=*device_ID*** command.

  The output shows the hardware acceleration, or VAAI, status that can be unknown, supported, or unsupported.

  ```
  # esxcli storage core device list -d naa.XXXXXXXXXXXX4c
  naa.XXXXXXXXXXXX4c
   Display Name: XXXX Fibre Channel Disk(naa.XXXXXXXXXXXX4c)
   Size: 20480
   Device Type: Direct-Access
   Multipath Plugin: NMP
   XXXXXXXXXXXXXXXX
   Attached Filters: VAAI_FILTER
   VAAI Status: supported
   XXXXXXXXXXXXXXXX
  ```

Verify Hardware Acceleration Support Details

Use the `esxcli` command to query whether the block storage device provides the hardware acceleration support.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- Run the **esxcli storage core device vaai status get -d=*device_ID*** command.

  If a VAAI plug-in manages the device, the output shows the name of the plug-in attached to the device. The output also shows the support status for each T10 SCSI based primitive, if available. Output appears in the following example:

```
# esxcli storage core device vaai status get -d naa.XXXXXXXXXXXX4c
naa.XXXXXXXXXXXX4c
 VAAI Plugin Name: VMW_VAAIP_SYMM
 ATS Status: supported
 Clone Status: supported
 Zero Status: supported
 Delete Status: unsupported
```

List Hardware Acceleration Claim Rules

Each block storage device managed by a VAAI plug-in needs two claim rules. One claim rule specifies the hardware acceleration filter and another specifies the hardware acceleration plug-in for the device. You can use the `esxcli` commands to list the hardware acceleration filter and plug-in claim rules.

Procedure

1 To list the filter claim rules, run the **esxcli storage core claimrule list --claimrule-class=Filter** command.

  In this example, the filter claim rules specify devices that the VAAI_FILTER filter claims.

```
# esxcli storage core claimrule list --claimrule-class=Filter
Rule Class   Rule   Class   Type   Plugin      Matches                    XCOPY Use Array
Reported Values  XCOPY Use Multiple Segments  XCOPY Max Transfer Size KiB
Filter       65430  runtime vendor VAAI_FILTER
vendor=EMC model=SYMMETRIX                           False
False                           0
Filter       65430  file    vendor VAAI_FILTER
vendor=EMC model=SYMMETRIX                           False
False                           0
Filter       65431  runtime vendor VAAI_FILTER
vendor=DGC model=*                                   False
```

```
False                           0
Filter       65431  file    vendor VAAI_FILTER
vendor=DGC model=*                                 False
False                           0
```

2 To list the VAAI plug-in claim rules, run the **esxcli storage core claimrule list --claimrule-class=VAAI** command.

   In this example, the VAAI claim rules specify devices that the VAAI plug-in claims.

```
esxcli storage core claimrule list --claimrule-class=VAAI
Rule Class   Rule   Class   Type    Plugin         Matches                  XCOPY Use
Array Reported Values  XCOPY Use Multiple Segments  XCOPY Max Transfer Size KiB
VAAI         65430  runtime vendor  VMW_VAAIP_SYMM
vendor=EMC model=SYMMETRIX                        False
False                           0
VAAI         65430  file    vendor  VMW_VAAIP_SYMM
vendor=EMC model=SYMMETRIX                        False
False                           0
VAAI         65431  runtime vendor  VMW_VAAIP_CX
vendor=DGC model=*                                False
False                           0
VAAI         65431  file    vendor  VMW_VAAIP_CX
vendor=DGC model=*                                False
False                           0
```

Add Hardware Acceleration Claim Rules

To configure the hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plug-in. For the new claim rules to be active, you first define the rules and then load them into your system.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Define a new claim rule for the VAAI filter by running the **esxcli storage core claimrule add --claimrule-class=Filter --plugin=VAAI_FILTER** command.

2 Define a new claim rule for the VAAI plug-in by running the **esxcli storage core claimrule add --claimrule-class=VAAI** command.

3 Load both claim rules by running the following commands:

   esxcli storage core claimrule load --claimrule-class=Filter

   esxcli storage core claimrule load --claimrule-class=VAAI

4 Run the VAAI filter claim rule by using the **esxcli storage core claimrule run --claimrule-class=Filter** command.

> **Note** Only the filter-class rules must be run. When the VAAI filter claims a device, it automatically finds the proper VAAI plug-in to attach.

Example: Defining Hardware Acceleration Claim Rules

This example shows how to configure the hardware acceleration for IBM arrays using the VMW_VAAIP_T10 plug-in. Use the following sequence of commands. For information about the options that the command takes, see Add Multipathing Claim Rules.

esxcli storage core claimrule add --claimrule-class=Filter --plugin=VAAI_FILTER --type=vendor --vendor=IBM --autoassign

esxcli storage core claimrule add --claimrule-class=VAAI --plugin=VMW_VAAIP_T10 --type=vendor --vendor=IBM --autoassign

esxcli storage core claimrule load --claimrule-class=Filter

esxcli storage core claimrule load --claimrule-class=VAAI

esxcli storage core claimrule run --claimrule-class=Filter

Configure XCOPY Parameters

XCOPY is one of the VAAI primitives that is used for offloading tasks to the storage array. For example, you can use XCOPY to offload such operations as migration or cloning of virtual machines to the array instead of consuming vSphere resources to perform these tasks.

You can use the XCOPY mechanism with all storage arrays that support the SCSI T10 based VMW_VAAIP_T10 plug-in developed by VMware. To enable the XCOPY mechanism, create a claim rule of the VAAI class.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

◆ Use the following command and enter the XCOPY options:

```
esxcli storage core claimrule add --claimrule-class=VAAI
```

For information about the options that the command takes, see Add Multipathing Claim Rules.

Option	Description
-a|--xcopy-use-array-values	Use array reported values for XCOPY commands.
-s|--xcopy-use-multi-segs	Use multiple segments for XCOPY commands. Valid only when `--xcopy-use-array-values` is specified.

Option	Description
-m|--xcopy-max-transfer-size	Maximum transfer size in MB for the XCOPY commands when you use a transfer size different than array reported. Valid only when `--xcopy-use-array-values` is specified.
-k|--xcopy-max-transfer-size-kib	Maximum transfer size in KiB for the XCOPY commands when you use a transfer size different than array reported. Valid only if `--xcopy-use-array-values` is specified.
-e | --extended-xcopy-support	Use the extended XCOPY command to optimize the data copy between datastores that are in different storage arrays.

Example: Configuring XCOPY

- ```
  # esxcli storage core claimrule add -r 914 -t vendor -V XtremIO -M XtremApp -P
  VMW_VAAIP_T10 -c VAAI -a -s -k 64
  ```

- ```
  # esxcli storage core claimrule add -r 65430 -t vendor -V EMC -M SYMMETRIX -P
  VMW_VAAIP_SYMM -c VAAI -a -s -m 200
  ```

Optimize Data Copying with Extended XCOPY Parameter

The extended XCOPY parameter optimizes the data copy between datastores across two different arrays. It allows you to migrate and clone workloads to offload XCOPY between two different arrays.

Extended XCOPY characteristics:

- Use a new extended XCOPY option in a claim rule.

  `esxcli storage core claimrule add -t vendor -V` *`vendor`* `-M` *`model name`* `-P VMW_VAAIP_T10 -c VAAI --autoassign -e`

  Example:

  `esxcli storage core claimrule add -t vendor -V DellEMC -M PowerStore -P VMW_VAAIP_T10 -c VAAI --autoassign -e`

- Extended XCOPY is supported only on T10 plug-in.

Procedure

1 Use the following command to check the extended XCOPY status.

  `esxcli storage core device vaai status get -d` *`device`*

  Example:

  ```
  [root@tpp-h127:~] esxcli storage core device vaai status get -d
  naa.68ccf098007f554c61148eb1786e6749
  naa.68ccf098007f554c61148eb1786e6749
     VAAI Plugin Name: VMW_VAAIP_T10
     ATS Status: supported
  ```

```
Clone Status: supported
Zero Status: supported
Delete Status: supported
Ex Clone Status: supported"
```

2 Enable extended XCOPY for the supported storage array.

a Verify if the VAAI_FILTER plug-in device is added.

```
esxcli storage core claimrule list -c Filter
```

b If the device is not added for VAAI_FILTER plug-in, use the following command.

```
esxcli storage core claimrule add -t vendor -V vendor -M model name -P
VAAI_FILTER -c Filter --autoassign
```

c Load the VAAI_FILTER claim rule.

```
esxcli storage core claimrule load -c Filter
```

d Add the extended XCOPY claim rule with T10 plug-in.

```
esxcli storage core claimrule add -t vendor -V vendor -M model name -P
VMW_VAAIP_T10 -c VAAI --autoassign -e
```

Note

- The claim rule in step d uses ESXi default transfer size and single segment for XCOPY commands. To use array reported values of different transfer size and multiple segments, use optional parameters *-a*, *-s*, *-m*/*-k* with the above command.
- Extended XCOPY setup works across different arrays.
- Add a claim rule with optional parameters such as *-a*, *-s*, *-m*/*-k* for both source and destination devices. In this case, the minimum transfer size for source and destination arrays will be considered for data transfer.

  Source Device:

  ```
  esxcli storage core claimrule add -t vendor -V vendor -M model name -P
  VMW_VAAIP_T10 -c VAAI --autoassign -a -s
  ```

  Destination Device:

  ```
  esxcli storage core claimrule add -t vendor -V vendor -M model name -P
  VMW_VAAIP_T10 -c VAAI --autoassign -a -s -e
  ```

e Load the VAAI_T10 plug-in.

```
esxcli storage core claimrule load -c VAAI
```

f Run the newly added claim rule options.

```
esxcli storage core claimrule run --claimrule-class=Filter
```

Delete Hardware Acceleration Claim Rules

Use the `esxcli` command to delete existing hardware acceleration claim rules.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- Run the following commands:

  `esxcli storage core claimrule remove -r claimrule_ID --claimrule-class=Filter`

  `esxcli storage core claimrule remove -r claimrule_ID --claimrule-class=VAAI`

vSphere Hardware Acceleration on NAS Devices

With the hardware acceleration, ESXi hosts can integrate with NAS devices and use several hardware operations, such as Full File Clone and Fast File Clone that NAS storage provides. The hardware acceleration uses vSphere APIs for Array Integration (VAAI) to facilitate communications between the hosts and storage devices.

The VAAI NAS framework supports both versions of NFS storage, NFS 3 and NFS 4.1.

The VAAI NAS uses a set of storage primitives to offload storage operations from the host to the array. The following list shows the supported NAS operations:

Full File Clone

Supports an ability of NAS device to clone virtual disk files. This operation is similar to the VMFS block cloning, except that NAS devices clone entire files instead of file segments. Tasks that benefit from the full file clone operation include VM cloning, Storage vMotion, and deployment of VMs from templates.

When the ESXi host copies data with VAAI NAS, it does not need to read the data from the NAS and write back the data to the NAS. The host simply sends the copy command offloading it to the NAS. The copy process is done in the NAS, which reduces the load on the host.

Fast File Clone

This operation, also called array-based or native snapshots, offloads creation of virtual machine snapshots and linked clones to the array.

Reserve Space

Supports an ability of storage arrays to allocate space for a virtual disk file in the thick format.

Typically, when you create a virtual disk on an NFS datastore, the NAS server determines the allocation policy. The default allocation policy on most NAS servers is thin and does not guarantee backing storage to the file. However, the reserve space operation can instruct the NAS device to use vendor-specific mechanisms to reserve space for a virtual disk. As a result, you can create thick virtual disks on the NFS datastore if the backing NAS server supports the reserve space operation.

Extended Statistics

Supports visibility to space use on NAS devices. The operation enables you to query space utilization details for virtual disks on NFS datastores. The details include the size of a virtual disk and the space consumption of the virtual disk. This functionality is useful for thin provisioning.

With NAS storage devices, the hardware acceleration integration is implemented through vendor-specific NAS plug-ins. These plug-ins are typically created by vendors and are distributed as vendor packages. No claim rules are required for the NAS plug-ins to function.

Several tools for installing and updating NAS plug-ins are available. They include the `esxcli` commands and vSphere Lifecycle Manager. For more information, see the *VMware ESXi Upgrade* and *Managing Host and Cluster Lifecycle* documentation. For installation and update recommendations, see the Knowledge Base article.

Note NAS storage vendors might provide additional settings that can affect the performance and operation of VAAI. Follow the vendor's recommendations and configure the appropriate settings on both the NAS storage array and ESXi. See your storage vendor documentation for more information.

Enable NAS Native Snapshots on Virtual Machines

If your deployment includes NAS arrays that support the vSphere APIs for Array Integration (VAAI), you can use the fast file clone technology, also called native NFS snapshots, to make virtual machine snapshots. With this technology, the NFS device copies the virtual machine without having the ESXi host read and write the data. This operation might reduce the time and network load when VM snapshots are created.

By default, all newly created VMs support traditional ESXi snapshot technology. To use the NFS native snapshot technology, enable it for the VM.

Prerequisites

- Verify that the NAS array supports the fast file clone operation with the VAAI NAS program.
- On your ESXi host, install vendor-specific NAS plug-in that supports the fast file cloning with VAAI.
- Follow the recommendations of your NAS storage vendor to configure any required settings on both the NAS array and ESXi. See your storage vendor documentation for more information.

Procedure

1 In the vSphere Client, right-click the virtual machine and select **Edit Settings**.

2 Click the **VM Options** tab and expand the **Advanced** menu.

3 Click **Edit Configuration** next to Configuration Parameters.

4 Configure the **snapshot.alwaysAllowNative** parameter.

If the parameter exists, make sure that its value is set to True. If the parameter is not present, add it and set its value to True.

Name	Value
snapshot.alwaysAllowNative	True

Note VMs stored on NFS datastores and configured with **snapshot.alwaysAllowNative** might encounter the `Invalid Disk Chain` error. For more information, see the VMware Knowledge Base article 89923.

Storage Provisioning and Space Reclamation in vSphere

17

vSphere supports two models of storage provisioning, thick provisioning and thin provisioning.

Thick provisioning

It is a traditional model of the storage provisioning. With the thick provisioning, large amount of storage space is provided in advance in anticipation of future storage needs. However, the space might remain unused causing underutilization of storage capacity.

Thin provisioning

This method contrast with thick provisioning and helps you eliminate storage underutilization problems by allocating storage space in a flexible on-demand manner. With ESXi, you can use two models of thin provisioning, array-level and virtual disk-level.

Thin provisioning allows you to report more virtual storage space than there is real physical capacity. This discrepancy can lead to storage over-subscription, also called over-provisioning. When you use thin provisioning, monitor actual storage usage to avoid conditions when you run out of physical storage space.

Read the following topics next:

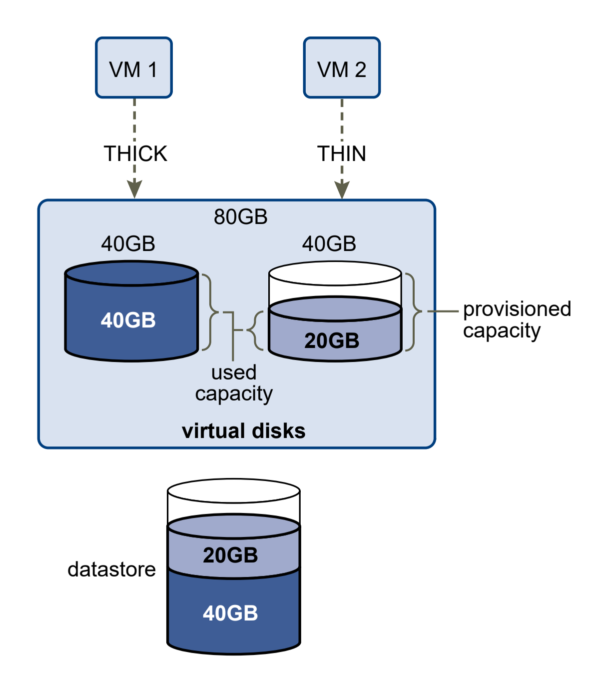
- Virtual Disk Thin Provisioning with vSphere Storage
- ESXi and Array Thin Provisioning
- Storage Space Reclamation in vSphere

Virtual Disk Thin Provisioning with vSphere Storage

When you create a virtual machine, a certain amount of storage space on a datastore is provisioned to virtual disk files. ESXi supports thick and thin provisioning for virtual disks.

By default, ESXi offers a traditional storage provisioning method for virtual machines. With this method, you first estimate how much storage the virtual machine might need for its entire life cycle. You then provision a fixed amount of storage space to the VM virtual disk in advance, for example, 40 GB. The entire provisioned space is committed to the virtual disk. A virtual disk that immediately occupies the entire provisioned space is a thick disk.

ESXi supports thin provisioning for virtual disks. With the disk-level thin provisioning feature, you can create virtual disks in a thin format. For a thin virtual disk, ESXi provisions the entire space required for the disk's current and future activities, for example 40 GB. However, the thin disk uses only as much storage space as the disk needs for its initial operations. In this example, the thin-provisioned disk occupies only 20 GB of storage. If the disk requires more space, it can expand into its entire 40 GB of provisioned space.



About Virtual Disk Provisioning Policies

When you perform certain virtual machine management operations, you can specify a provisioning policy for the virtual disk file. The operations include creating a virtual disk, cloning a virtual machine to a template, or migrating a virtual machine.

NFS datastores with Hardware Acceleration and VMFS datastores support the following disk provisioning policies. On NFS datastores that do not support Hardware Acceleration, only thin format is available.

You can use Storage vMotion or cross-host Storage vMotion to transform virtual disks from one format to another.

Thick Provision Lazy Zeroed

Creates a virtual disk in a default thick format. Space required for the virtual disk is allocated when the disk is created. Data remaining on the physical device is not erased during creation, but is zeroed out on demand later on first write from the virtual machine. Virtual machines do not read stale data from the physical device.

Thick Provision Eager Zeroed

A type of thick virtual disk that supports clustering features such as Fault Tolerance. Space required for the virtual disk is allocated at creation time. In contrast to the thick provision lazy zeroed format, the data remaining on the physical device is zeroed out when the virtual disk is created. It might take longer to create virtual disks in this format than to create other types of disks. Increasing the size of an Eager Zeroed Thick virtual disk causes a significant stun time for the virtual machine.

Thin Provision

Use this format to save storage space. For the thin disk, you provision as much datastore space as the disk would require based on the value that you enter for the virtual disk size. However, the thin disk starts small and at first, uses only as much datastore space as the disk needs for its initial operations. If the thin disk needs more space later, it can grow to its maximum capacity and occupy the entire datastore space provisioned to it.

Thin provisioning is the fastest method to create a virtual disk because it creates a disk with just the header information. It does not allocate or zero out storage blocks. Storage blocks are allocated and zeroed out when they are first accessed.

Note If a virtual disk supports clustering solutions such as Fault Tolerance, do not make the disk thin.

Create Thin Provisioned Virtual Disks

To save storage space, you can create a virtual disk in a thin provisioned format. The thin provisioned virtual disk starts small and expands as more disk space is required. You can create thin disks only on the datastores that support disk-level thin provisioning.

This procedure assumes that you are creating a new virtual machine. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

1 Create a virtual machine.

   a Right-click any inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host, and select **New Virtual Machine**.

   b Select **Create a new virtual machine** and click **Next**.

   c Follow the steps required to create a virtual machine.

2 Configure the thin virtual disk.

   a On the Customize Hardware page, click the **Virtual Hardware** tab.

   b Click the **New Hard disk** triangle to expand the hard disk options.

c (Optional) Adjust the default disk size.

With a thin virtual disk, the disk size value shows how much space is provisioned and guaranteed to the disk. At the beginning, the virtual disk might not use the entire provisioned space. The actual storage use value can be less than the size of the virtual disk.

d Select **Thin Provision** for Disk Provisioning.

3 Finish the virtual machine creation.

Results

You created a virtual machine with a disk in the thin format.

What to do next

If you created a virtual disk in the thin format, you can later inflate it to its full size.

View Virtual Machine Storage Resources

You can view how datastore storage space is allocated for your virtual machines.

Procedure

1 Browse to the virtual machine.

2 Double-click the virtual machine and click the **Summary** tab.

3 Review the storage use information in the upper right area of the **Summary** tab.

Results

Storage Usage shows how much datastore space is occupied by virtual machine files, including configuration and log files, snapshots, virtual disks, and so on. When the virtual machine is running, the used storage space also includes swap files.

For virtual machines with thin disks, the actual storage use value might be less than the size of the virtual disk.

Determine the Disk Format of a Virtual Machine

You can determine whether your virtual disk is in thick or thin format.

Procedure

1 Right-click the virtual machine and select **Edit Settings**.

2 Click the **Virtual Hardware** tab.

3 Click the **Hard Disk** triangle to expand the hard disk options.

The **Type** text box shows the format of your virtual disk.

What to do next

If your virtual disk is in the thin format, you can inflate it to its full size.

Inflate Thin Virtual Disks

If you created a virtual disk in the thin format, you can change the format to thick.

You use the datastore browser to inflate the thin virtual disk.

Prerequisites

- Make sure that the datastore where the virtual machine resides has enough space.
- Make sure that the virtual disk is thin.
- Remove snapshots.
- Power off your virtual machine.

Procedure

1 In the vSphere Client, navigate to the folder of the virtual disk you want to inflate.

   a Navigate to the virtual machine.

   b Click the **Datastores** tab.

   The datastore that stores the virtual machine files is listed.

   c Right-click the datastore and select **Browse Files**.

   The datastore browser displays contents of the datastore.

2 Expand the virtual machine folder and browse to the virtual disk file that you want to convert.

   The file has the `.vmdk` extension and is marked with the virtual disk () icon.

3 Select the virtual disk file and click **Inflate**.

   Note The option might not be available if the virtual disk is thick or when the virtual machine is running.

Results

The inflated virtual disk occupies the entire datastore space originally provisioned to it.

Handling Datastore Over-Subscription

Because the provisioned space for thin disks can be greater than the committed space, a datastore over-subscription can occur, which results in the total provisioned space for the virtual machine disks on the datastore being greater than the actual capacity.

Over-subscription can be possible because usually not all virtual machines with thin disks need the entire provisioned datastore space simultaneously. However, if you want to avoid over-subscribing the datastore, you can set up an alarm that notifies you when the provisioned space reaches a certain threshold.

For information on setting alarms, see the *vCenter Server and Host Management* documentation.

If your virtual machines require more space, the datastore space is allocated on a first come first served basis. When the datastore runs out of space, you can add more physical storage and increase the datastore.

See Increase Capacity of vSphere VMFS Datastore .

ESXi and Array Thin Provisioning

You can use thin-provisioned storage arrays with ESXi.

The ESXi host integrates with block-based storage and performs these tasks:

- The host can recognize underlying thin-provisioned LUNs and monitor their space use to avoid running out of physical space. The LUN space might change if, for example, your VMFS datastore expands or if you use Storage vMotion to migrate virtual machines to the thin-provisioned LUN. The host warns you about breaches in physical LUN space and about out-of-space conditions.
- The host can run the automatic T10 `unmap` command from VMFS6 and VM guest operating systems to reclaim unused space from the array. VMFS5 supports a manual space reclamation method.

Note ESXi does not support activating or deactivating of thin provisioning on a storage device.

Requirements

To use the thin provisioning reporting and space reclamation features, follow these requirements:

- Use an appropriate ESXi version.

Supported thin provisioning components	ESXi 6.5 and later
Thin provisioning	Yes
Unmap command originating from VMFS	Automatic for VMFS6
Unmap command originating from guest OS	Yes (VMFS6)

- Use storage systems that support T10-based vSphere Storage APIs - Array Integration (VAAI), including thin provisioning and space reclamation. For information, contact your storage provider and check the *VMware Compatibility Guide* documentation.

Monitoring Space Use

The thin provision integration functionality helps you to monitor the use of space on thin-provisioned LUNs and to avoid running out of space.

The following sample flow demonstrates how the ESXi host and the storage array interact to generate breach of space and out-of-space warnings for a thin-provisioned LUN. The same mechanism applies when you use Storage vMotion to migrate virtual machines to the thin-provisioned LUN.

1. Using storage-specific tools, your storage administrator provisions a thin LUN and sets a soft threshold limit that, when reached, triggers an alert. This step is vendor-specific.
2. Using the vSphere Client, you create a VMFS datastore on the thin-provisioned LUN. The datastore spans the entire logical size that the LUN reports.
3. As the space used by the datastore increases and reaches the set soft threshold, the following actions take place:
   a. The storage array reports the breach to your host.
   b. Your host triggers a warning alarm for the datastore.

      You can contact the storage administrator to request more physical space. Alternatively, you can use Storage vMotion to evacuate your virtual machines before the LUN runs out of capacity.
4. If no space is left to allocate to the thin-provisioned LUN, the following actions take place:
   a. The storage array reports out-of-space condition to your host.

      Caution In certain cases, when a LUN becomes full, it might go offline or get unmapped from the host.

   b. The host pauses virtual machines and generates an out-of-space alarm.

      You can resolve the permanent out-of-space condition by requesting more physical space from the storage administrator.

Identify Thin-Provisioned Storage Devices

Use the `esxcli` command to verify whether a particular storage device is thin-provisioned.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- Run the **esxcli storage core device list -d=*device_ID*** command.

Results

The following thin provisioning status indicates that the storage device is thin-provisioned.

```
# esxcli storage core device list -d naa.XXXXXXXXXXXX4c
naa.XXXXXXXXXXXX4c
 Display Name: XXXX Fibre Channel Disk(naa.XXXXXXXXXXXX4c)
 Size: 20480
 Device Type: Direct-Access
 Multipath Plugin: NMP
 ---------------------
 Thin Provisioning Status: yes
 ---------------------
```

An unknown status indicates that a storage device is thick.

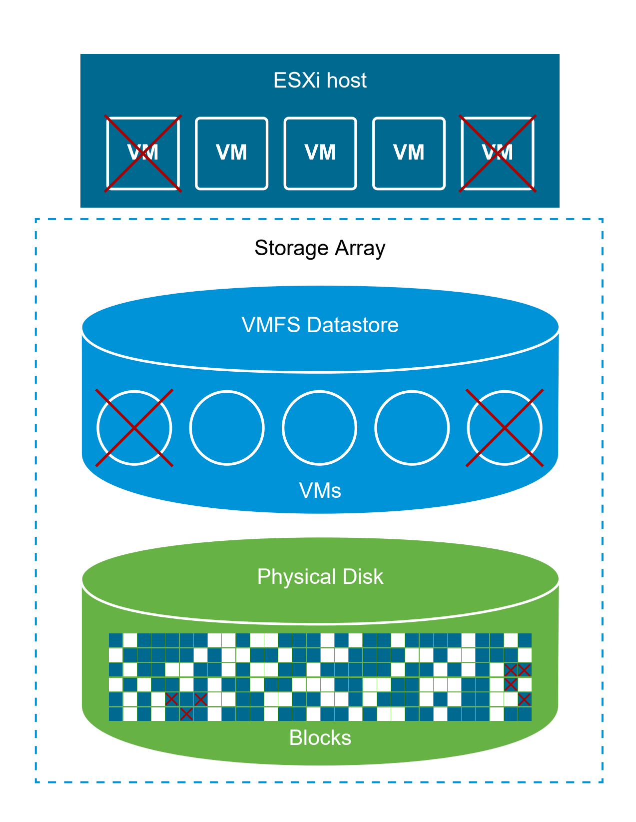
Note Some storage systems present all devices as thin-provisioned no matter whether the devices are thin or thick. Their thin provisioning status is always `yes`. For details, check with your storage vendor.

Storage Space Reclamation in vSphere

ESXi supports the space reclamation command, also called SCSI unmap command or NVMe dataset management deallocate command, that originates from a VMFS datastore or a VM guest operating system. The command helps thin-provisioned storage arrays to reclaim unused space from the VMFS datastore and thin virtual disks on the datastore. The VMFS6 datastore can send the space reclamation command automatically. With the VMFS5 datastore, you can manually reclaim storage space.

For information about space reclamation with the Virtual Volumes datastores, see Reclaim Space on the vSphere Virtual Volumes Datastores.

You free storage space inside the VMFS datastore when you delete or migrate the VM, consolidate a snapshot, and so on. Inside the virtual machine, storage space is freed when you delete files on the thin virtual disk. These operations leave blocks of unused space on the storage array. However, when the array is not aware that the data was deleted from the blocks, the blocks remain allocated by the array until the datastore releases them. VMFS uses the SCSI unmap command to indicate to the array that the storage blocks contain deleted data, so that the array can unallocate these blocks.



The command can also originate directly from the guest operating system. Both VMFS5 and VMFS6 datastores can provide support to the unmap command that proceeds from the guest operating system. However, the level of support is limited on VMFS5.

Depending on the type of your VMFS datastore, you use different methods to configure space reclamation for the datastore and your virtual machines.

Watch the following video to learn more about how space reclamation works.

 (Space Reclamation with VMFS)

Space Reclamation on vSphere VMFS Datastores

Deleting or removing files from a VMFS datastore frees space within the file system. This free space is mapped to a storage device until the file system releases or unmaps it. ESXi supports reclamation of free space, which is also called the unmap operation.

The operation helps the storage array to reclaim unused free space. Unmapped space can be then used for other storage allocation requests and needs.

Automatic Reclamation of Free Space on VMFS6 Datastore

On VMFS6 datastores, ESXi supports the automatic asynchronous reclamation of free space. VMFS6 can run the unmap command to release free storage space in the background on thin-provisioned storage arrays that support unmap operations.

Automatic unmap processing has several advantages:

- Unmap requests are sent at a constant rate, which helps to avoid any instant load on the backing array.
- Freed regions are batched and unmapped together.
- I/O performance of other workloads is not impacted by the unmap command.

For VMFS6 datastores, you can configure the following space reclamation parameters.

Space reclamation granularity

Granularity defines the minimum size of a released space sector that underlying storage can reclaim. Storage cannot reclaim those sectors that are smaller in size than the specified granularity.

For VMFS6, reclamation granularity equals the block size. When you specify the block size as 1 MB, the granularity is also 1 MB. Storage sectors of the size smaller than 1 MB are not reclaimed.

Note Certain storage arrays recommend an optimal unmap granularity. ESXi supports automatic unmap processing on arrays with the recommended unmap granularity of 1 MB or greater, for example, 16 MB. On the arrays with the optimal granularity of 1 MB and less, the unmap operation is supported if the granularity is a factor of 1 MB. For example, 1 MB is divisible by 512 bytes, 4 KB, 64 KB, and so on.

Space reclamation method

The method can be either priority or fixed. When the method you use is priority, you configure the priority rate. For the fixed method, you must indicate the reclamation rate in MB per second.

You can use the vSphere Client to set the priority method when you create the VMFS6 datastore. To enable the fixed method, you can later modify the space reclamation settings for the existing datastore.

Alternately, you can use the `esxcli storage vmfs reclaim config set` command to set space reclamation parameters.

Space reclamation priority rate

This parameter defines the rate at which the space reclamation operation is performed when you use the priority reclamation method. Typically, VMFS6 can send the unmap commands either in bursts or sporadically depending on the workload and configuration. For VMFS6, you can specify one of the following options.

Space Reclamation Priority	Description	Configuration
None	Deactivates the unmap operations for the datastore.	vSphere Client `esxcli` command
Low (default)	Sends the unmap command at a less frequent rate, 25–50 MB per second.	vSphere Client `esxcli` command
Medium	Sends the command at a rate twice faster than the low rate, 50–100 MB per second.	`esxcli` command
High	Sends the command at a rate three times faster than the low rate, over 100 MB per second.	`esxcli` command

Space reclamation hosts

This parameter defines the number of ESXi hosts that can issue unmap commands to a VMFS6 datastore for automatic space reclamation. Use this parameter to limit the cumulative amount of space reclamation load issued to the target backing VMFS6 datastore.

After you enable space reclamation, the VMFS6 datastore can start releasing the blocks of unused space only when it has at least one open file. This condition can be fulfilled when, for example, you power on one of the VMs on the datastore.

Configure Priority Reclamation when Creating a VMFS6 Datastore

Space reclamation settings control how deleted or unmapped blocks are reclaimed on the devices that back the datastore. When you create a VMFS6 datastore in the vSphere Client, the only method for space reclamation you can specify is the priority method. You can also deactivate space reclamation when creating the datastore.

Procedure

1. In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
2. From the right-click menu, select **Storage > New Datastore**.
3. Follow the steps required to create a VMFS6 datastore.

4 On the **Partition configuration** page, specify the space reclamation parameters.

The parameters define granularity and the priority rate at which space reclamation operations are performed. You can also use this page to deactivate space reclamation for the datastore.

Option	Description
Block size	The block size on a VMFS datastore defines the maximum file size and the amount of space the file occupies. VMFS6 supports the block size of 1 MB.
Space reclamation granularity	Specify granularity for the unmap operation. Unmap granularity equals the block size, which is 1 MB. Storage sectors of the size smaller than 1 MB are not reclaimed.
Space reclamation priority	Select one of the following options. ■ Low (default). Use the priority method for space reclamation. The unmap operation is performed at a low priority rate. ■ None. Select this option if you do not want to use the space reclamation operations for the datastore.

Note In the vSphere Client, the only available settings for the space reclamation priority are Low and None. To change the settings to Medium or High, use the `esxcli` command.

5 Finish the datastore creation process.

Results

After you activate space reclamation, the VMFS6 datastore can start releasing the blocks of unused space only when it has at least one open file. This condition can be fulfilled when, for example, you power on one of the VMs on the datastore.

Configure Fixed Reclamation for the Existing VMFS6 Datastore

To activate the fixed reclamation method, modify the space reclamation settings for the datastore.

Procedure

1 In the vSphere Client, navigate to the datastore.

2 Select **Edit Space Reclamation** from the right-click menu.

3 Specify the space reclamation setting.

Option	Description
Enable automatic space reclamation at fixed rate	Use the fixed method for space reclamation. Specify reclamation rate in MB per second. Minimum available value is 10 MB per second.
Disable automatic space reclamation	Deleted or unmapped blocks are not reclaimed. Select this option if you want to deactivate the space reclamation operations for the datastore.

4 Click **OK** to save the new settings.

5 Unmount and remount the datastore for the changes to take place.

   a Unmount Datastores.

   b Mount Datastores.

6 Repeat this procedure for all ESXi hosts that access the datastore.

Results

The modified value for the space reclamation priority appears on the **General** page for the datastore.

Use the ESXCLI Command to Change Space Reclamation on VMFS6

You can change the default space reclamation priority, granularity, and other parameters.

Procedure

1 On the ESXi host, use the following command to set space reclamation parameters.

   `esxcli storage vmfs reclaim config set`

   The command takes these options:

Option	Description
-b|--reclaim-bandwidth	Space reclamation fixed bandwidth in MB per second.
-g|--reclaim-granularity	Minimum granularity of automatic space reclamation in bytes.
-m|--reclaim-method	Method of automatic space reclamation. Supported options: ■ priority ■ fixed
-n|--reclaim-maxhosts	Maximum number of ESXi hosts that can issue unmap commands to a VMFS6 datastore for automatic space reclamation.
-p|--reclaim-priority	Priority of automatic space reclamation. Supported options: ■ none ■ low ■ medium ■ high
-l|--volume-label	The label of the target VMFS volume.
-u|--volume-uuid	The uuid of the target VMFS volume.

   You can use the following examples.

   - Set the reclamation method to fixed and the rate to 100 MB per second.

```
esxcli storage vmfs reclaim config set --volume-label datastore_name --reclaim-method
fixed -b 100
```

- Deactivate the VMFS automatic space reclamation.

```
esxcli storage vmfs reclaim config set --volume-label datastore_name --reclaim-
priority none
```

Note If you set only the `-n|--reclaim-maxhosts` parameter without any other parameters, you can skip Step 2. It takes around 30 minutes for the updated reclaim-maxhosts parameter value to take effect on all ESXi hosts where the VMFS6 datastore mounted.

2 Unmount the VMFS6 datastore from all other ESXi hosts where the datastore is mounted, and then remount it again.

 a Unmount Datastores.

 b Mount Datastores.

 This step ensures that all ESXi hosts where the VMFS6 datastore was mounted update the reclamation settings for the datastore.

Verify Settings for Automatic Space Reclamation on VMFS6

After you configure or edit space reclamation parameters for a VMFS6 datastore, you can review your settings.

Procedure

1 In the vSphere Client, navigate to the datastore.

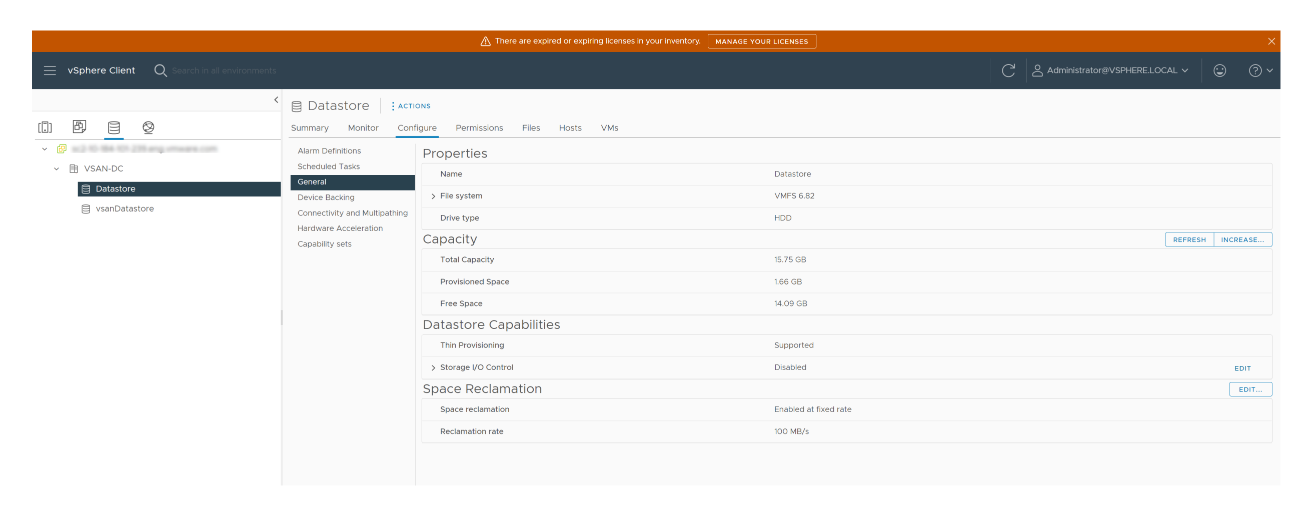
2 Click the **Configure** tab.

3 Click **General**.

4 Under **Space Reclamation**, review the setting for the space reclamation priority.

 If you configured any values through the `esxcli` command, for example, Medium or High for the space reclamation priority, these values also appear in the vSphere Client.

Results



Example: Obtaining Parameters for VMFS6 Space Reclamation

You can also use the **esxcli storage vmfs reclaim config get -l=*VMFS_label*|-u=*VMFS_uuid*** command to obtain information for the space reclamation configuration.

```
# esxcli storage vmfs reclaim config get -l my_datastore
Reclaim Granularity: 1048576 Bytes
Reclaim Priority: low
Reclaim Method: priority
Reclaim Bandwidth: 26 MB/s
Reclaim Maximum Hosts: 128
```

Manually Reclaim Accumulated Storage Space on VMFS5

On VMFS datastores that do not support automatic space reclamation, such as VMFS5, you can use the `esxcli` command to reclaim unused storage space manually.

When you use the command, be aware that it might send many unmap requests at a time. This action can lock some of the resources during the operation. When you use the command, be aware that it might send many unmap requests at a time. This action can lock some of the resources during the operation.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 To reclaim unused storage blocks on the thin-provisioned device, run the following command:

esxcli storage vmfs unmap

The command takes these options:

Option	Description
-l|--volume-label=*volume_label*	The label of the VMFS volume to unmap. A mandatory argument. If you specify this argument, do not use **-u|--volume-uuid=*volume_uuid***.
-u|--volume-uuid=*volume_uuid*	The UUID of the VMFS volume to unmap. A mandatory argument. If you specify this argument, do not use **-l|--volume-label=*volume_label***.
-n|--reclaim-unit=*number*	Number of VMFS blocks to unmap per iteration. An optional argument. If it is not specified, the command uses the default value of 200.

2 To verify whether the unmap process has finished, search for unmap in the `vmkernel.log` file.

Reclaim Space on the vSphere Virtual Volumes Datastores

When your ESXi host uses the SCSI or NVMe protocol to connect to storage in the Virtual Volumes environment, the data and configuration virtual volumes are formatted with VMFS6. As a result, the space reclamation process available for VMFS can also be applied to both the data

and configuration Virtual Volumes. When you reclaim space on the Virtual Volumes datastore, you can use the `esxcli storage vvol` command to manually unmap free blocks from the virtual volumes.

Note Starting with 8.0 Update 3:

- NVMe backed data Virtual Volumes support space reclamation requests from guest operating systems.
- Configuration Virtual Volumes support automatic space reclamation. To reclaim the space manually, follow the steps in the given procedure.

Prerequisites

- For information about virtual volumes, see Types of Virtual Volumes.
- For information about the space reclamation process, see Storage Space Reclamation in vSphere.

Procedure

1 Obtain the storage container ID and meta vVol UUID for your configuration virtual volume formatted with VMFS.

   a Navigate to the VM that contains virtual volumes.

      For example:

      ```
      [root@:~] cd /vmfs/volumes/vvol_datastore/VM_1
      ```

   b From the path, extract the storage container ID and meta vVol UUID.

      ```
      [root@:/vmfs/volumes/vvol:4abe7eb4bf234ee0xxxxxxxxx/
      rfc4122.1ffee4bd-9c55-49b9-876d-xxxxxxxx
      ```

      In this example,

      Container ID is `vvol:4abe7eb4bf234ee0xxxxxxxxx`

      Meta vVol UUID is `rfc4122.1ffee4bd-9c55-49b9-876d-xxxxxxxx`

2 Use the following command to unmap free blocks from the configuration volume.

   ```
   esxcli storage vvol virtualvolume unmap
   ```

   The command takes the following options:

Option	Description
`-c|--container-id=<str>`	Storage container ID. The option is required.
`-n|--reclaim-unit=<long>`	Number of VMFS blocks to unmap per iteration.
`-u|--uuid=<str>`	Meta vVol UUID. The option is required.

3 Verify that you successfully unmapped the blocks by checking the `hostd` log file.

   ```
   grep "Total Unmapped blocks from vmfs" /var/run/log/hostd.log
   ```

Example:

1.
```
[root@:~]esxcli storage vvol virtualvolume unmap -c vvol:4abe7eb4bf234ee0xxxxxxxxx -u
rfc4122.1ffee4bd-9c55-49b9-876d-xxxxxxxx
```

2.
```
[root@:~] grep "Total Unmapped blocks from vmfs" /var/run/log/hostd.log
```

   The output looks similar to the following:

```
2023-06-01T10:27:45.836Z In(166) Hostd[1000342556]: [Originator@6876
sub=Libs opID=esxcli-5f-1861 sid=523fac9d user=root] Unmap: Done : Total
Unmapped blocks from vmfs6 volume rfc4122.1ffee4bd-9c55-49b9-87 6d-
xxxxxxxx : 0 (LFB Pass)

2023-06-01T10:28:13.279Z In(166) Hostd[1000342556]: [Originator@6876
sub=Libs opID=esxcli-5f-1861 sid=523fac9d user=root] Unmap: Done : Total
Unmapped blocks from vmfs6 volume rfc4122.1ffee4bd-9c55-49b9-87 6d-
xxxxxxxx : 259421 (SFB Pass)
```

Space Reclamation Requests from Guest Operating Systems

ESXi supports the unmap commands issued directly from a guest operating system to reclaim storage space. The level of support and requirements depend on the type of datastore where your virtual machine resides.

Inside a virtual machine, storage space is freed when, for example, you delete files on the thin virtual disk. The guest operating system notifies VMFS about freed space by sending the unmap command. The unmap command sent from the guest operating system releases space within the VMFS datastore. The command then proceeds to the array, so that the array can reclaim the freed blocks of space.

Space Reclamation for VMFS6 Virtual Machines

VMFS6 generally supports automatic space reclamation requests that generate from the guest operating systems, and passes these requests to the array. Many guest operating systems can send the unmap command and do not require any additional configuration. The guest operating systems that do not support the automatic unmaps might require user intervention. For information about guest operating systems that support the automatic space reclamation on VMFS6, contact your vendor.

Generally, the guest operating systems send the unmap commands based on the unmap granularity they advertise. For details, see documentation provided with your guest operating system.

The following considerations apply when you use space reclamation with VMFS6:

- VMFS6 processes the unmap request from the guest OS only when the space to reclaim equals 1 MB or is a multiple of 1 MB. If the space is less than 1 MB or is not aligned to 1 MB, the unmap requests are not processed.

- For VMs with snapshots in the default SEsparse format, VMFS6 supports the automatic space reclamation only on ESXi hosts version 6.7 or later.

  Space reclamation affects only the top snapshot and works when the VM is powered on.

Space Reclamation for VMFS5 Virtual Machines

Typically, the unmap command that generates from the guest operation system on VMFS5 cannot be passed directly to the array. You must run the `esxcli storage vmfs unmap` command to trigger unmaps for the array.

However, for a limited number of the guest operating systems, VMFS5 supports the automatic space reclamation requests.

To send the unmap requests from the guest operating system to the array, the virtual machine must meet the following prerequisites:

- The virtual disk must be thin-provisioned.
- Virtual machine hardware must be of version 11 (ESXi 6.0) or later.
- The advanced setting EnableBlockDelete must be set to 1.
- The guest operating system must be able to identify the virtual disk as thin.

Using vmkfstools in vSphere

18

`vmkfstools` is one of the ESXi Shell commands for managing VMFS volumes, storage devices, and virtual disks. You can perform many storage operations using the `vmkfstools` command. For example, you can create and manage VMFS datastores on a physical partition, or manipulate virtual disk files, stored on VMFS or NFS datastores.

Note After you make a change using the `vmkfstools`, the vSphere Client might not be updated immediately. Use a refresh or rescan operation from the client.

For more information on the ESXi Shell, see *Getting Started with ESXCLI*.

vmkfstools Command Syntax

Generally, you do not need to log in as the root user to run the `vmkfstools` commands. However, some commands, such as the file system commands, might require the root user login.

The `vmkfstools` command supports the following command syntax:

`vmkfstools` *options target*.

Target specifies a partition, device, or path to apply the command option to.

Table 18-1. `vmkfstools` Command Arguments

Argument	Description
options	One or more command-line options and associated arguments that you use to specify the activity for `vmkfstools` to perform. For example, selecting the disk format when creating a new virtual disk. After entering the option, specify a target on which to perform the operation. Target can indicate a partition, device, or path.
partition	Specifies disk partitions. This argument uses a *disk_ID:P* format, where *disk_ID* is the device ID returned by the storage array and *P* is an integer that represents the partition number. The partition digit must be greater than zero (0) and must correspond to a valid VMFS partition.

Table 18-1. `vmkfstools` Command Arguments (continued)

Argument	Description
device	Specifies devices or logical volumes. This argument uses a path name in the ESXi device file system. The path name begins with `/vmfs/devices`, which is the mount point of the device file system. Use the following formats when you specify different types of devices: ■ `/vmfs/devices/disks` for local or SAN-based disks. ■ `/vmfs/devices/lvm` for ESXi logical volumes. ■ `/vmfs/devices/generic` for generic SCSI devices.

path	Specifies a VMFS file system or file. This argument is an absolute or relative path that names a directory symbolic link, a raw device mapping, or a file under `/vmfs`.
	■ To specify a VMFS file system, use this format:
	`/vmfs/volumes/file_system_UUID`
	or
	`/vmfs/volumes/file_system_label`
	■ To specify a file on a VMFS datastore, use this format:
	`/vmfs/volumes/file_system_label|file_system_UUID/[dir]/myDisk.vmdk`
	If the current working directory is the parent directory of `myDisk.vmdk`, do not enter the entire path.

The vmkfstools Command Options

The `vmkfstools` command has several options. Some of the options are suggested for advanced users only.

The long and single-letter forms of the options are equivalent. For example, the following commands are identical.

```
vmkfstools --createfs vmfs6 --blocksize 1m disk_ID:P
vmkfstools -C vmfs6 -b 1m disk_ID:P
```

-v Suboption

The `-v` suboption indicates the verbosity level of the command output.

The format for this suboption is as follows:

```
-v --verbose number
```

You specify the *number* value as an integer from 1 through 10.

You can specify the `-v` suboption with any `vmkfstools` option. If the output of the option is not suitable for use with the `-v` suboption, `vmkfstools` ignores `-v`.

Note Because you can include the `-v` suboption in any `vmkfstools` command line, `-v` is not included as a suboption in the option descriptions.

Read the following topics next:

- File System Options of vSphere vmkfstools Command
- Virtual Disk Options
- Storage Device Options of vSphere vmkfstools Command

File System Options of vSphere vmkfstools Command

File system options allow you to create and manage VMFS datastores. These options do not apply to NFS. You can perform many of these tasks through the vSphere Client.

Listing Attributes of a VMFS Datastore

Use the `vmkfstools` command to list attributes of a VMFS datastore.

```
-P|--queryfs
      -h|--humanreadable
```

When you use this option on any file or directory that resides on a VMFS datastore, the option lists the attributes of the specified datastore. The listed attributes typically include the file system label, the number of extents for the datastore, the UUID, and a list of the devices where each extent resides.

Note If any device backing VMFS file system goes offline, the number of extents and available space change accordingly.

You can specify the `-h|--humanreadable` suboption with the `-P` option. If you do so, `vmkfstools` lists the capacity of the volume in a more readable form.

Example: Example of Listing VMFS Attributes

```
~ vmkfstools -P -h /vmfs/volumes/my_vmfs
VMFS-5.81 (Raw Major Version: 14) file system spanning 1 partitions.
File system label (if any): my_vmfs
Mode: public
Capacity 99.8 GB, 97.5 GB available, file block size 1 MB, max supported file size 62.9 TB
UUID: 571fe2fb-ec4b8d6c-d375-XXXXXXXXXXXX
Partitions spanned (on "lvm"):
        eui.3863316131XXXXXX:1
Is Native Snapshot Capable: YES
```

Creating a VMFS Datastore or a Scratch Partition

Use the `vmkfstools` command to create a VMFS datastore or a scratch partition.

```
-C|--createfs [vmfs5|vmfs6|vfat]
```

This option creates the VMFS datastore on the specified SCSI or NVMe partition, such as *disk_ID:P*. The partition becomes the head partition of the datastore. For VMFS5 and VMFS6, the only available block size is 1 MB.

You can specify the following suboptions with the `-C` option.

- `-S|--setfsname` - Define the volume label of the VMFS datastore you are creating. Use this suboption only with the `-C` option. The label you specify can be up to 128 characters long and cannot contain any leading or trailing blank spaces.

  Note vCenter Server supports the 80 character limit for all its entities. If a datastore name exceeds this limit, the name gets shortened when you add this datastore to vCenter Server.

  After you define a volume label, you can use it whenever you specify the VMFS datastore for the `vmkfstools` command. The volume label appears in listings generated for the `ls -l` command and as a symbolic link to the VMFS volume under the `/vmfs/volumes` directory.

  To change the VMFS volume label, use the `ln -sf` command. Use the following as an example:

  ```
  ln -sf /vmfs/volumes/UUID /vmfs/volumes/datastore
  ```

  datastore is the new volume label to use for the *UUID* VMFS.

  Note If your host is registered with vCenter Server, any changes you make to the VMFS volume label get overwritten by vCenter Server. This operation guarantees that the VMFS label is consistent across all vCenter Server hosts.

- `-Y|--unmapGranularity #[bBsSkKmMgGtT]` - This suboption applies to VMFS6 only. Define granularity for the unmap operation. The default granularity is 1 MB. As with the block size, enter the unit type.
- `-O|--unmapPriority <none|low|medium|high>` - This suboption applies to VMFS6 only. Define the priority for the unmap operation.

Example: Example for Creating a VMFS File System

This example illustrates creating a VMFS6 datastore named my_vmfs on the `naa.ID:1` partition.

```
~ vmkfstools -C vmfs6 -S my_vmfs /vmfs/devices/disks/naa.ID:1
```

Adding an Extent to a VMFS Datastore

Use the `vmkfstools` command to add an extent to a VMFS datastore.

When you add an extent, you span the VMFS datastore from the head partition across the partition specified by *span_partition*.

```
-Z|--spanfs span_partition head_partition
```

You must specify the full path name for the head and span partitions, for example `/vmfs/devices/disks/disk_ID:1`. Each time you use this option, you add an extent to the VMFS datastore, so that the datastore spans multiple partitions.

Caution When you run this option, you lose all data that previously existed on the SCSI or NVMe device you specified in *span_partition*.

Example: Example for Extending a VMFS Datastore

In this example, you extend the existing head partition of the VMFS datastore over a new partition.

```
~ vmkfstools -Z /vmfs/devices/disks/naa.disk_ID_2:1 /vmfs/devices/disks/naa.disk_ID_1:1
```

The extended datastore spans two partitions, `naa.disk_ID_1:1` and `naa.disk_ID_2:1`. In this example, `naa.disk_ID_1:1` is the name of the head partition.

Expanding a VMFS Datastore

Instead of adding an extent to a VMFS datastore, you can increase the size of an existing datastore. Use the `vmkfstools -G` command.

You might increase the datastore size after the underlying storage had its capacity increased.

The command uses the following option:

```
-G|--growfs device device
```

This option expands the VMFS datastore or its specific extent. For example,

```
vmkfstools --growfs /vmfs/devices/disks/disk_ID:1 /vmfs/devices/disks/disk_ID:1
```

Virtual Disk Options

Virtual disk options allow you to set up, migrate, and manage virtual disks stored on your datastores. You can also perform most of these tasks through the vSphere Client.

Supported Disk Formats

When you create or clone a virtual disk, you can use the `-d|--diskformat` suboption to specify the format for the disk.

Choose from the following formats:

- zeroedthick (default) – Space required for the virtual disk is allocated during creation. Any data remaining on the physical device is not erased during creation, but is zeroed out on demand on first write from the virtual machine. The virtual machine does not read stale data from disk.
- eagerzeroedthick – Space required for the virtual disk is allocated at creation time. In contrast to zeroedthick format, the data remaining on the physical device is zeroed out during creation. It might take much longer to create disks in this format than to create other types of disks.
- thin – Thin-provisioned virtual disk. Unlike with the thick format, space required for the virtual disk is not allocated during creation, but is supplied, zeroed out, on demand.
- rdm:*device* – Virtual compatibility mode raw disk mapping.
- rdmp:*device* – Physical compatibility mode (pass-through) raw disk mapping.
- 2gbsparse – A sparse disk with the maximum extent size of 2 GB. You can use disks in this format with hosted VMware products, such as VMware Fusion. However, you cannot power on the sparse disk on an ESXi host unless you first re-import the disk with vmkfstools in a compatible format, such as thick or thin.

Disk Formats on NFS Datastores

The only disk formats you can use for NFS are thin, thick, zeroedthick, and 2gbsparse.

Thick, zeroedthick, and thin formats usually behave the same because the NFS server and not the ESXi host determines the allocation policy. The default allocation policy on most NFS servers is thin. However, on NFS servers that support Storage APIs - Array Integration, you can create virtual disks in zeroedthick format. The reserve space operation enables NFS servers to allocate and guarantee space.

For more information on array integration APIs, see Chapter 16 Storage Hardware Acceleration in vSphere.

Creating a Virtual Disk

Use the vmkfstools command to create a virtual disk.

```
-c|--createvirtualdisk size[bB|sS|kK|mM|gG]
      -d|--diskformat [thin|zeroedthick|eagerzeroedthick]
      -W|--objecttype [file|vsan|vvol]
      --policyFile fileName
```

This option creates a virtual disk at the specified path on a datastore. Specify the size of the virtual disk. When you enter the value for *size*, you can indicate the unit type by adding a suffix of k (kilobytes), m (megabytes), or g (gigabytes). The unit type is not case-sensitive. vmkfstools interprets either k or K to mean kilobytes. If you do not specify a unit type, vmkfstools defaults to bytes.

You can specify the following suboptions with the `-c` option.

- `-d|--diskformat` specifies disk formats.
- `-W|--objecttype` specifies whether the virtual disk is a file on a VMFS or NFS datastore, or an object on a vSAN or Virtual Volumes datastore.
- `--policyFile` *`fileName`* specifies VM storage policy for the disk.

Example: Example for Creating a Virtual Disk

This example shows how to create a two-gigabyte virtual disk file named `disk.vmdk`. You create the disk on the VMFS datastore named `myVMFS`. The disk file represents an empty virtual disk that virtual machines can access.

```
vmkfstools -c 2048m /vmfs/volumes/myVMFS/disk.vmdk
```

Initializing a Virtual Disk

Use the `vmkfstools` command to initialize a virtual disk.

```
-w|--writezeros
```

This option cleans the virtual disk by writing zeros over all its data. Depending on the size of your virtual disk and the I/O bandwidth to the device hosting the virtual disk, completing this command might take a long time.

Note This command is not supported on NFS datastores.

Caution When you use this command, you lose any existing data on the virtual disk.

Inflating a Thin Virtual Disk

Use the `vmkfstools` command to inflate a thin virtual disk.

```
-j|--inflatedisk
```

This option converts a `thin` virtual disk to `eagerzeroedthick`, preserving all existing data. The option allocates and zeroes out any blocks that are not already allocated.

Converting a Zeroedthick Virtual Disk to an Eagerzeroedthick Disk

Use the `vmkfstools` command to convert any zeroedthick virtual disk to an eagerzeroedthick disk.

```
-k|--eagerzero
```

While performing the conversion, this option preserves any data on the virtual disk.

Follow this example:

```
vmkfstools --eagerzero /vmfs/volumes/myVMFS/VMName/disk.vmdk
```

Removing Zeroed Blocks

Use the `vmkfstools` command to remove zeroed blocks.

```
-K|--punchzero
```

This option deallocates all zeroed out blocks and leaves only those blocks that were allocated previously and contain valid data. The resulting virtual disk is in thin format.

Deleting a Virtual Disk

Use the vmkfstools command to delete a virtual disk file at the specified path on the VMFS volume.

Use the following option:

```
-U|--deletevirtualdisk
```

Renaming a Virtual Disk

Use the vmkfstools command to rename a virtual disk file at the specified path on the VMFS volume.

You must specify the original filename or file path *oldName* and the new filename or file path *newName*.

```
-E|--renamevirtualdisk oldName newName
```

Cloning or Converting a Virtual Disk or RDM

Use the vmkfstools command to create a copy of a virtual disk or raw disk you specify.

A non-root user cannot clone a virtual disk or an RDM. You must specify the original filename or file path *oldName* and the new filename or file path *newName*.

```
 -i|--clonevirtualdisk oldName newName
      -d|--diskformat [thin|zeroedthick|eagerzeroedthick|rdm:device|rdmp:device|2gbsparse]
      -W|--objecttype [file|vsan|vvol]
      --policyFile fileName
      -N|--avoidnativeclone
```

Use the following suboptions to change corresponding parameters for the copy you create.

- `-d|--diskformat` specifies disk formats.

- `-W|--objecttype` specifies whether the virtual disk is a file on a VMFS or NFS datastore, or an object on a vSAN or Virtual Volumes datastore.
- `--policyFile` *`fileName`* specifies VM storage policy for the disk.

By default, ESXi uses its native methods to perform the cloning operations. If your array supports the cloning technologies, you can off-load the operations to the array. To avoid the ESXi native cloning, specify the `-N|--avoidnativeclone` option.

Example: Example for Cloning or Converting a Virtual Disk

This example illustrates cloning the contents of a master virtual disk from the `templates` repository to a virtual disk file named `myOS.vmdk` on the `myVMFS` file system.

```
vmkfstools -i /vmfs/volumes/myVMFS/templates/gold-master.vmdk /vmfs/volumes/myVMFS/myOS.vmdk
```

You can configure a virtual machine to use this virtual disk by adding lines to the virtual machine configuration file, as in the following example:

```
scsi0:0.present = TRUE
scsi0:0.fileName = /vmfs/volumes/myVMFS/myOS.vmdk
```

If you want to convert the format of the disk, use the `-d|--diskformat` suboption.

This suboption is useful when you import virtual disks in a format not compatible with ESXi, for example 2gbsparse format. After you convert the disk, you can attach this disk to a new virtual machine you create in ESXi.

For example:

```
vmkfstools -i /vmfs/volumes/myVMFS/templates/gold-master.vmdk /vmfs/volumes/myVMFS/myOS.vmdk
-d thin
```

Extending a Virtual Disk

After you create a virtual machine, you can use the vmkfstools command to extend the size of a disk allocated to the virtual machine.

```
-X|--extendvirtualdisk newSize[bBsSkKmMgGtT]
```

Specify the `newSize` parameter adding an appropriate unit suffix. The unit type is not case-sensitive. `vmkfstools` interprets either `k` or `K` to mean kilobytes. If you do not specify the unit type, `vmkfstools` defaults to kilobytes.

The `newSize` parameter defines the entire new size, not just the increment you add to the disk.

For example, to extend a 4-g virtual disk by 1 g, enter: `vmkfstools -X 5g` *disk name*.

You can extend the virtual disk to the eagerzeroedthick format by using the `-d eagerzeroedthick` option.

When you use the -X option, the following considerations apply:

- Do not extend the base disk of a virtual machine that has snapshots associated with it. If you do, you can no longer commit the snapshot or revert the base disk to its original size.
- After you extend the disk, you might need to update the file system on the disk. As a result, the guest operating system recognizes the new size of the disk and can use it.

Upgrading Virtual Disks

This option converts the specified virtual disk file from ESX Server 2 formats to the ESXi format.

Use this option to convert virtual disks of type LEGACYSPARSE, LEGACYPLAIN, LEGACYVMFS, LEGACYVMFS_SPARSE, and LEGACYVMFS_RDM.

```
-M|--migratevirtualdisk
```

Creating a Virtual Compatibility Mode Raw Device Mapping

Use the vmkfstools command to create a Raw Device Mapping (RDM) file on a VMFS volume and map a raw LUN to this file. After this mapping is established, you can access the LUN as you would a normal VMFS virtual disk. The file length of the mapping is the same as the size of the raw LUN it points to.

```
-r|--createrdm device
```

When specifying the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

Example: Example for Creating a Virtual Compatibility Mode RDM

In this example, you create an RDM file named *my_rdm.vmdk* and map the *disk_ID* raw disk to that file.

```
vmkfstools -r /vmfs/devices/disks/disk_ID my_rdm.vmdk
```

You can configure a virtual machine to use the `my_rdm.vmdk` mapping file by adding the following lines to the virtual machine configuration file:

```
scsi0:0.present = TRUE
scsi0:0.fileName = /vmfs/volumes/myVMFS/my_rdm.vmdk
```

Creating a Physical Compatibility Mode Raw Device Mapping

Use the vmkfstools command to map a pass-through raw device to a file on a VMFS volume. With the mapping, a virtual machine can bypass ESXi SCSI command filtering when accessing its

virtual disk. This type of mapping is useful when the virtual machine must send proprietary SCSI commands, for example, when SAN-aware software runs on the virtual machine.

```
-z|--createrdmpassthru device example.vmdk
```

After you establish this type of mapping, you can use it to access the raw disk as you access any other VMFS virtual disk.

When specifying the *device* path, use the following format:

```
/vmfs/devices/disks/device_ID
```

For the .vmdk name, use this format. Make sure to create the datastore before using the command.

```
/vmfs/volumes/datastore_name/example.vmdk
```

For example,

```
vmkfstools -z /vmfs/devices/disks/naa.600a0000000000000... /vmfs/volumes/datastore1/
mydisk.vmdk
```

Listing Attributes of an RDM

Use the vmkfstools command to list the attributes of a raw disk mapping. The attributes help you identify the storage device to which your RDM files maps.

```
-q|--queryrdm my_rdm.vmdk
```

This option prints the name of the raw disk RDM. The option also prints other identification information, like the disk ID, for the raw disk.

Example: Example of Listing RDM Attributes

```
# vmkfstools -q /vmfs/volumes/VMFS/my_vm/my_rdm.vmdk

Disk /vmfs/volumes/VMFS/my_vm/my_rdm.vmdk is a Passthrough Raw Device Mapping

Maps to: vml.020000000060050768019002077000000000000005323134352020
```

Displaying Virtual Disk Geometry

Use the vmkfstools command to get information about the geometry of a virtual disk.

```
-g|--geometry
```

The output is in the form: `Geometry information C/H/S`, where `C` represents the number of cylinders, `H` represents the number of heads, and `S` represents the number of sectors.

Note When you import virtual disks from hosted VMware products to the ESXi host, you might see a disk geometry mismatch error message. A disk geometry mismatch might also trigger problems when you load a guest operating system or run a newly created virtual machine.

Checking and Repairing Virtual Disks

Use the vmkfstools command to check or repair a virtual disk if it gets corrupted.

```
-x|--fix [check|repair]
```

For example,

```
vmkfstools -x check /vmfs/volumes/my_datastore/my_disk.vmdk
```

Checking Disk Chain for Consistency

Use the vmkfstools command to check the entire snapshot chain. You can determine if any of the links in the chain are corrupted or any invalid parent-child relationships exist.

```
-e|--chainConsistent
```

Storage Device Options of vSphere vmkfstools Command

You can use the device options of the vSphere vmkfstools command to perform administrative task for physical storage devices.

Managing SCSI Reservations of LUNs

Use the vmkfstools command to reserve a SCSI LUN for exclusive use by the ESXi host. You can also release a reservation so that other hosts can access the LUN, and reset a reservation, forcing all reservations from the target to be released.

```
-L|--lock [reserve|release|lunreset|targetreset|busreset|readkeys|readresv] device
```

Caution Using the `-L` option can interrupt the operations of other servers on a SAN. Use the `-L` option only when troubleshooting clustering setups.

Unless advised by VMware, never use this option on a LUN hosting a VMFS volume.

You can specify the `-L` option in several ways:

- `-L reserve` – Reserves the specified LUN. After the reservation, only the server that reserved that LUN can access it. If other servers attempt to access that LUN, a reservation error appears.

- `-L release` – Releases the reservation on the specified LUN. Other servers can access the LUN again.
- `-L lunreset` – Resets the specified LUN by clearing any reservation on the LUN and making the LUN available to all servers again. The reset does not affect any of the other LUNs on the device. If another LUN on the device is reserved, it remains reserved.
- `-L targetreset` – Resets the entire target. The reset clears any reservations on all the LUNs associated with that target and makes the LUNs available to all servers again.
- `-L busreset` – Resets all accessible targets on the bus. The reset clears any reservation on all the LUNs accessible through the bus and makes them available to all servers again.
- `-L readkeys` – Reads the reservation keys registered with a LUN. Applies to SCSI-III persistent group reservation functionality.
- `-L readresv` – Reads the reservation state on a LUN. Applies to SCSI-III persistent group reservation functionality.

When entering the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

Breaking Device Locks

Use the vmkfstools command to break the device lock on a particular partition.

```
-B|--breaklock device
```

When entering the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

You can use this command when a host fails in the middle of a datastore operation, such as expand the datastore, add an extent, or resignature. When you run this command, make sure that no other host is holding the lock.

Getting Started with Cloud Native Storage in vSphere 19

Cloud Native Storage is a solution that provides comprehensive data management for stateful applications in vSphere. When you use Cloud Native Storage, you can create containerized stateful applications capable of surviving restarts and outages. Stateful containers leverage storage exposed by vSphere while using such primitives as standard volume, persistent volume, and dynamic provisioning.

With Cloud Native Storage, you can create persistent container volumes independent of virtual machine and container life cycle. vSphere storage backs the volumes, and you can set a storage policy directly on the volumes. After you create the volumes, you can review them and their backing storage objects in the vSphere Client, and monitor their storage policy compliance.

vSphere Cloud Native Storage supports persistent volumes in the following Kubernetes distributions:

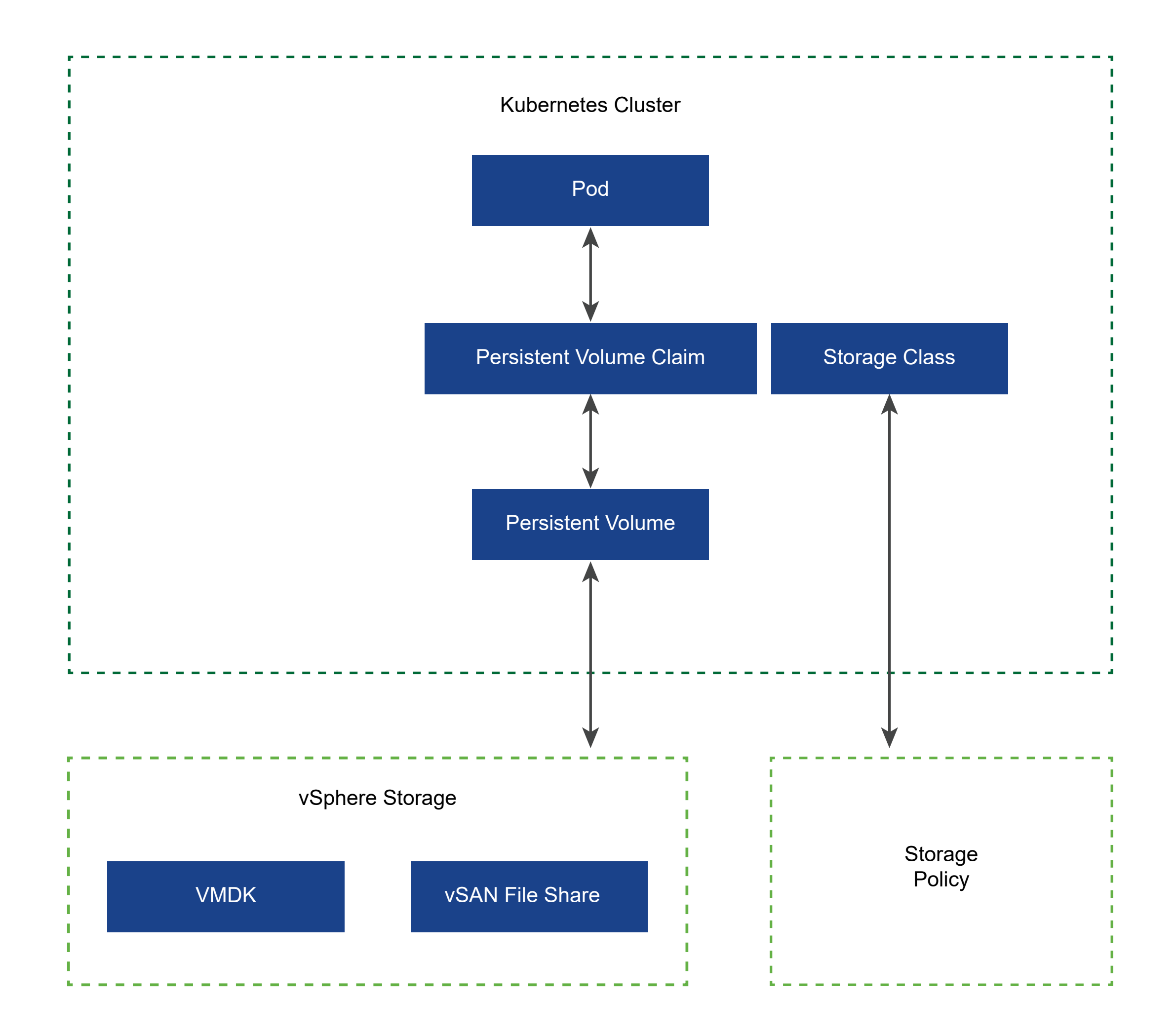
- Generic Kubernetes, also called vanilla, that you install from the official repositories. This *vSphere Storage* documentation covers only generic Kubernetes.
- vSphere IaaS control plane, formerly called vSphere with Tanzu. For more information, see the *vSphere IaaS Control Plane Concepts and Planning* documentation.

Read the following topics next:

- Cloud Native Storage Concepts and Terminology in vSphere
- Cloud Native Storage for vSphere Administrators

Cloud Native Storage Concepts and Terminology in vSphere

Be familiar with several concepts essential to the Cloud Native Storage environment in vSphere.



Kubernetes Cluster

In the Cloud Native Storage environment, you can deploy a generic Kubernetes cluster in a cluster of virtual machines. On top of the Kubernetes cluster, you deploy your containerized applications. Applications can be stateful and stateless.

Note For information on a Supervisor and TKG cluster that you can run in vSphere IaaS control plane environment, see the *vSphere IaaS Control Plane Concepts and Planning* documentation.

Pod

A pod is a group of one or more containerized applications that share such resources as storage and network. Containers inside a pod are started, stopped, and replicated as a group.

Container Orchestrator

Open-source platforms, such as Kubernetes, for deployment, scaling, and management of containerized applications across clusters of hosts. The platforms provide a container-centric infrastructure.

Stateful Application

As containerized applications evolve from stateless to stateful, they require persistent storage. Unlike stateless applications that do not save data between sessions, stateful applications save data to persistent storage. The retained data is called the application's state. You can later retrieve the data and use it in the next session. Most applications are stateful. A database is as an example of a stateful application.

PersistentVolume

Stateful applications use PersistentVolumes to store their data. A PersistentVolume is a Kubernetes volume capable of retaining its state and data. It is independent of a pod and can continue to exist even when the pod is deleted or reconfigured. In the vSphere environment, the PersistentVolume objects use vSphere virtual disks of the First Class Disk (FCD) type or vSAN file shares as their backing storage. First Class Disks are also referred to as Improved Virtual Disks (IVD) or managed virtual disks.

- Virtual disks support volumes that are mounted as ReadWriteOnce. These volumes can be used only by a single Pod in Kubernetes.

  You can use the vSphere encryption technology to protect FCD virtual disks that back persistent volumes. For more information, see Use Encryption with vSphere Cloud Native Storage.
- vSAN file shares support ReadWriteMany volumes that are mounted by many nodes. These volumes can be shared between multiple Pods or applications running across Kubernetes nodes or across Kubernetes clusters. For information about possible configurations with file shares, see Using vSAN File Service to Provision vSphere File Volumes.

StorageClass

Kubernetes uses a StorageClass to define different tiers of storage and to describe different types of requirements for storage backing the PersistentVolume. In the vSphere environment, a storage class can be linked to a storage policy. As a vSphere administrator, you create storage policies that describe different storage requirements. The VM storage policies can be used as a part of StorageClass definition for dynamic volume provisioning.

The following sample YAML file references the **`Gold`** storage policy that you created earlier using the vSphere Client. The resulting persistent volume VMDK is placed on a compatible datastore that satisfies the **`Gold`** storage policy requirements.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: gold-sc
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.vsphere.vmware.com
parameters:
```

```
  storagepolicyname: "Gold"
```

PersistentVolumeClaim

Typically, applications or pods can request persistent storage through a PersistentVolumeClaim. The PersistentVolumeClaim specifies the type and class of storage, the access mode, either ReadWriteOnce or ReadWriteMany, and other parameters for the PersistentVolume. The request can then dynamically provision the corresponding PersistentVolume object and the underlying virtual disk or vSAN file share in the vSphere environment.

Once the claim is created, the PersistentVolume is automatically bound to the claim. Pods use the claim to mount the PersistentVolume and access storage.

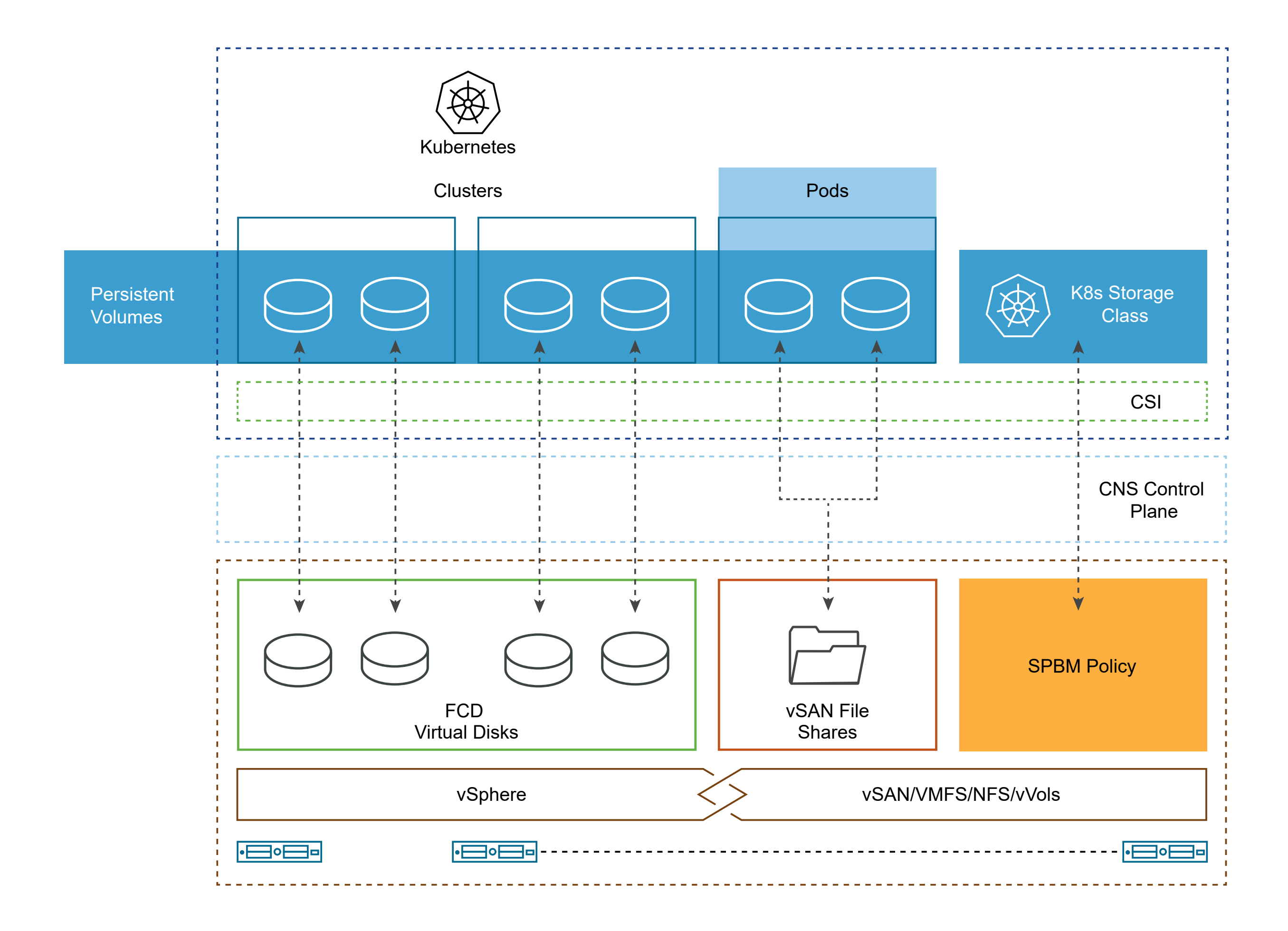
When you delete this claim, the corresponding PersistentVolume object and the underlying storage are deleted.

```
kind: PersistentVolumeClaim
metadata:
 name: persistent-VMDK
spec:
 accessModes:
 - ReadWriteOnce
 resources:
 requests:
 storage: 5Gi
 storageClassName: gold-sc
```

vSphere Cloud Native Storage Components

Cloud Native Storage uses several components to integrate with vSphere storage.

The following illustration shows how these components interact.



Kubernetes Cluster

In the Cloud Native Storage environment, a generic Kubernetes cluster is deployed on a cluster of virtual machines, or nodes, that run in vSphere. A Kubernetes user directly interacts with the cluster when deploying stateful applications on top of it.

Note For information on a Supervisor and TKG cluster that you can run in vSphere IaaS control plane environment, see the *vSphere IaaS Control Plane Concepts and Planning* documentation.

Container Storage Interface (CSI) for vSphere

To consume underlying infrastructure resources, the cluster requires a CSI driver.

The vSphere CSI is an out-of-tree plug-in that exposes vSphere storage to containerized workloads on container orchestrators, such as Kubernetes. The plug-in enables vSAN and other types of vSphere storage.

The vSphere CSI communicates with the CNS control plane on vCenter Server for all storage provisioning operations. The vSphere CSI supports the following functionalities:

- Dynamic provisioning of container volumes.
- The vSphere First Class Disk functionality.

- Kubernetes zones.
- Conventional and raw mounts.
- Single vCenter Server, and multiple data centers and clusters.
- Provisioning from multiple datastores or datastore clusters.
- vSAN File Service

On Kubernetes, the CSI driver is used with the out-of-tree vSphere Cloud Provider Interface (CPI). The CSI driver is shipped as a container image and must be deployed by the cluster administrator. For information, see the Driver Deployment section of the Kubernetes vSphere CSI Driver documentation.

For information about the CSI variations used in a Supervisor and TKG clusters, see the *vSphere IaaS Control Plane Concepts and Planning* documentation.

Cloud Native Storage Server Component

The CNS server component, or the CNS control plane, resides in vCenter Server. It is an extension of vCenter Server management that implements the provisioning and life cycle operations for the container volumes.

When provisioning container volumes, it interacts with vCenter Server to create storage objects that back the volumes. The Storage Policy Based Management functionality guarantees a required level of service to the volumes.

The CNS also performs query operations that allow you to manage and monitor container volumes and their backing storage objects through vCenter Server.

First Class Disk (FCD)

Also called Improved Virtual Disk (IVD) or managed virtual disk. It is a named virtual disk unassociated with a VM. These disks reside on a vSAN, VMFS, NFS, or vVols datastore and back ReadWriteOnce container volumes.

The FCD technology allows to perform life cycle operations related to persistent volumes outside of the VM or pod life cycle. If the VM is a Kubernetes node that runs multiple container based applications and uses persistent volumes and virtual disks for many applications, CNS facilitates life cycle operations at the container and persistent volume granularity.

vSAN File Service

It is a vSAN layer that provides file shares. Currently, it supports NFSv3 and NFSv4.1 file shares. Cloud Native Storage uses vSAN file shares for persistent volumes of the ReadWriteMany type. A single ReadWriteMany volume can be mounted by multiple nodes. The volume can be shared between multiple pods or applications running across Kubernetes nodes or across Kubernetes clusters.

Storage Policy Based Management

Storage Policy Based Management is a vCenter Server service that supports provisioning of persistent volumes according to specified storage requirements. After provisioning, the service monitors compliance of the volume with the required policy characteristics.

Using vSAN File Service to Provision vSphere File Volumes

vSAN file service offers vSAN file shares that are consumed by persistent volumes of the ReadWriteMany (RWM) type. A single RWM volume can be mounted by multiple nodes. The volume can be shared between multiple pods or applications running across Kubernetes nodes or across Kubernetes clusters in vSphere.

When a Kubernetes pod request an RWM volume, Cloud Native Storage communicates with vSAN file service to create an NFS-based file share of the requested size and storage class. Cloud Native Storage then mounts the RWM volume into the Kubernetes worker node where the pod runs. If multiple nodes are requesting access to the RWM volume, Cloud Native Storage determines that the RWM volume already exists for that particular deployment and mounts the existing volume into the nodes.

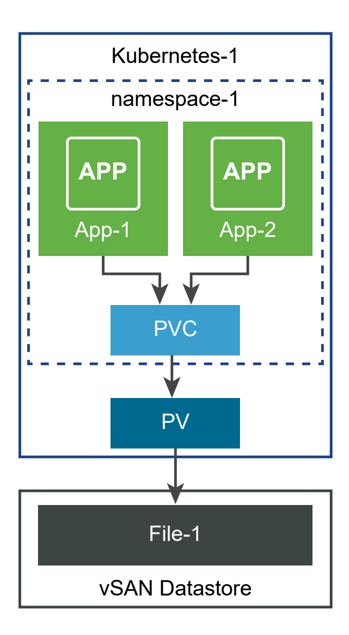
To be able to support RWM volumes, your environment must include the following items.

- vSphere 7.0 and later with vSAN
- vSAN file service enabled. For information, see vSAN File Service in the *Administering VMware vSAN* documentation.
- Kubernetes version 1.14 and later
- Compatible version of CSI. For information, see the VMware vSphere Container Storage Plug-in Documentation documentation.

You can use different configurations for file volumes.

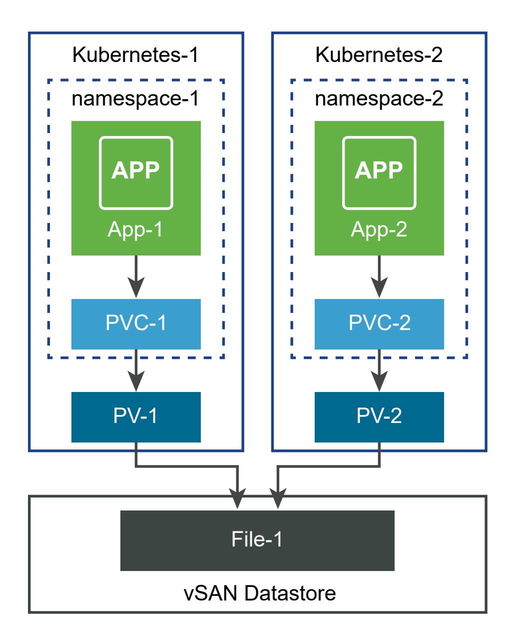
Single File Volume Shared Across Applications in the Same Namespace

In this example, a single file volume is used as shared storage across different applications in the same namespace. You use a single persistent volume claim to provision the file volume.



Single File Volume Shared Across Applications and Namespaces

This example uses a single file volume as shared storage across different applications and different namespaces. For each namespace, you create a separate persistent volume claim to provision the same file volume.



vSphere Cloud Native Storage Users

The types of users involved in the process of creating and monitoring Kubernetes volumes in the vSphere Cloud Native Storage environment generally fall into two categories, a Kubernetes user and a vSphere administrator. Both types of users have access to different tools and perform different tasks.

CNS Kubernetes User

The Kubernetes user might be a Kubernetes developer and an application owner, a Kubernetes administrator, or combine functions of both. The tasks that the Kubernetes user performs in the Cloud Native Storage environment include the following:

- Deploy and manage the vSphere CSI. For information, see the vSphere Container Storage Plug-in Deployment section of the Getting Started with VMware vSphere Container Storage Plug-in documentation.
- Provision persistent volumes. For information about block volumes, see vSphere CSI Driver - Block Volume. For information about file volumes, see vSphere CSI Driver - File Volume.
- Perform life cycle operations for persistent volumes.
- Perform life cycle operations for storage classes.

CNS vSphere User

A CNS vSphere user, or a vSphere administrator, has access to the vSphere Client to perform the following tasks:

- Perform life cycle operations for the VM storage policies. For example, create a VM storage policy to be used for a Kubernetes storage class and communicate its name to the Kubernetes user. See Create a Storage Policy for Kubernetes in vSphere.
- Use the Cloud Native Storage section of the vSphere Client to monitor health and storage policy compliance of the container volumes across the Kubernetes clusters. See Monitor Container Volumes Across Kubernetes Clusters in vSphere.

Cloud Native Storage for vSphere Administrators

A vSphere administrator delivers storage resources to the Kubernetes team and creates VM storage policies that describe different storage requirements and classes of services. After the Kubernetes workloads with persistent storage are provisioned, the vSphere administrator can monitor the life cycle of the backing storage resources and their compliance to the requirements.

Requirements of vSphere Cloud Native Storage

Your Cloud Native Storage environment and virtual machines that participate in the Kubernetes cluster must meet several requirements.

Cloud Native Storage Requirements

- A compatible version of vSphere.
- A compatible version of Kubernetes.
- A Kubernetes cluster deployed on the virtual machines. For details about deploying the vSphere CSI plug-in and running the Kubernetes cluster on vSphere, see the VMware vSphere Container Storage Plug-in Documentation.

Requirements for Kubernetes Cluster Virtual Machines

- Virtual machines with hardware version 15 or later. Install VMware Tools on each node virtual machine.
- Virtual machine hardware recommendations:
  - Set CPU and memory adequately based on workload requirements.
  - Use the VMware Paravirtual SCSI controller for the primary disk on the Node VM.
- All virtual machines must have access to a shared datastore, such as vSAN.
- Set the `disk.EnableUUID` parameter on each node VM. See Configure Kubernetes Cluster in vSphere Virtual Machines.
- To avoid errors and unpredictable behavior, do not take snapshots of CNS node VMs.

Requirements for CNS File Volume

- Use vSphere version 7.0 or later with a compatible Kubernetes version.
- Use a compatible version of CSI. For information, see the VMware vSphere Container Storage Plug-in Documentation.
- Enable and configure the vSAN file service. You must configure the necessary file service domains, IP pools, network, and so on. For information, see the *Administering VMware vSAN* documentation.
- Follow specific guidelines to configure network access from a guest OS in the Kubernetes node to a vSAN file share. See Configuring Network Access to vSAN File Share.

Configuring Network Access to vSAN File Share

To be able to provision ReadWriteMany persistent volumes in your generic vSphere Kubernetes environment, configure necessary networks, switches, and routers from the Kubernetes nodes to the vSAN file service network.

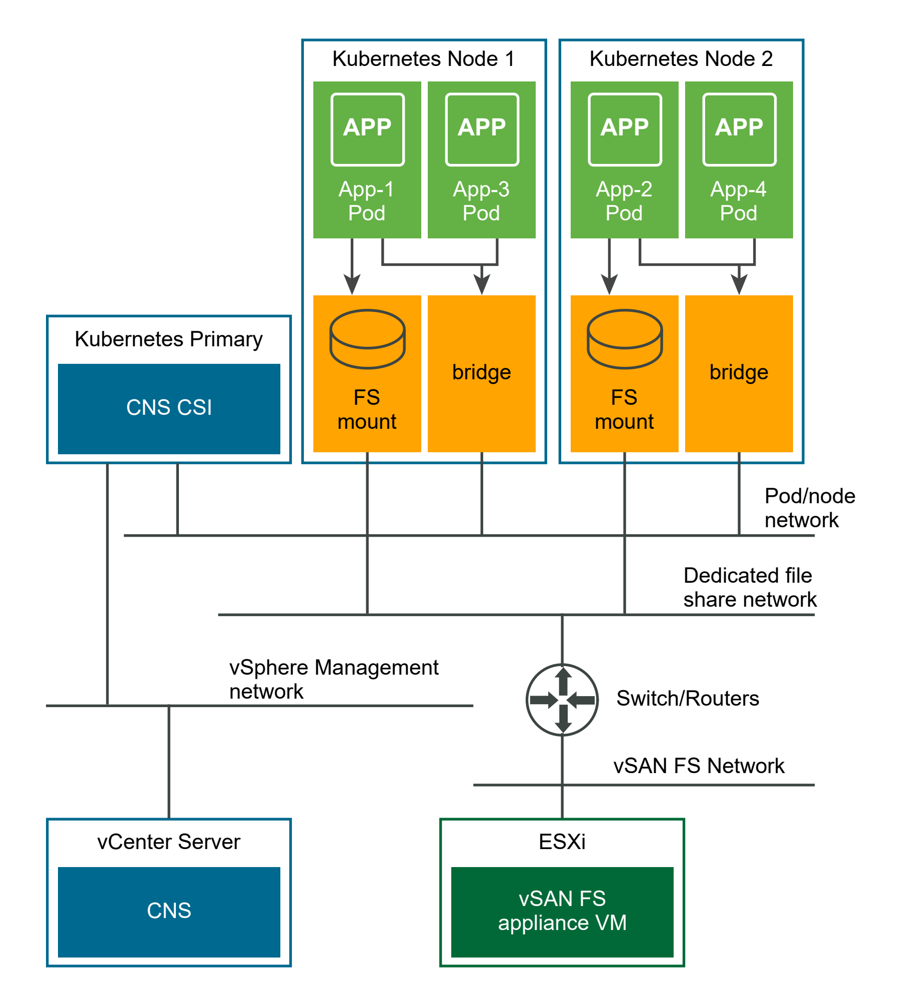
Setting Up Network

When configuring the networks, follow these requirements:

- On every Kubernetes node, you can use a dedicated vNIC for the vSAN file share traffic. This option is required only if you want to use a secure data traffic path for your file volumes.
- If you use a dedicated vNIC, make sure that the traffic through the dedicated vNIC is routable to one or many vSAN file service networks.
- Make sure that only the guest OS on each Kubernetes node can directly access the vSAN file share through the file share IP address. The pods in the node cannot ping or access the vSAN file share by its IP address.

  CNS CSI driver ensures that only those pods that are configured to use the CNS file volume can access the vSAN file share by creating a mount point in the guest OS.
- Avoid creating an IP address clash between the node VMs and vSAN file shares.

The following illustration is an example of the CNS network configuration with the vSAN file share service.



In the illustration, the sample networking configuration follows these guidelines.

- The configuration uses separate networks for different items in the CNS environment.

Network	Description
vSphere management network	Typically, in a generic Kubernetes cluster, every node has access to this network.
Pod or node network	Kubernetes uses this network for the node to node or pod to pod communication.
Dedicated file share network	CNS file volume data traffic uses this network.
vSAN file share network	Network where the vSAN file share is enabled and where file shares are available.

- Every Kubernetes node has a dedicated vNIC for the file traffic. This vNIC is separate from the vNIC used for the node to node or pod to pod communication. This configuration is used only as an example, but is not mandatory.

- Only those applications that are configured to use the CNS file share have access to vSAN file shares through the mount point in the node guest OS. For example, in the illustration, the following takes place:
  - App-1 and App-2 pods are configured to use a file volume, and have access to the file share through the mount point created by the CSI driver.
  - App-3 and App-4 are not configured with a file volume and cannot access file shares.
- The vSAN file shares are deployed as containers in a vSAN file share appliance VM on the ESXi host. A Kubernetes deployer, which is a software or service that can configure, deploy, and manage Kubernetes clusters, configures necessary routers and switches, so that the guest OS in the Kubernetes node can access the vSAN file shares.

Security Limitations

Although the dedicated vNIC prevents an unauthorized pod from accessing the file shares directly, certain security limitations exist:

- The CNS file functionality assumes that anyone who has the CNS file volume ID is an authorized user of the volume. Any user that has the CNS file volume ID can access the data stored in the volume.
- CNS file volume supports only the AUTH_SYS authentication, which is a user ID-based authentication. To protect access to the data in the CNS file volume, you must use appropriate user IDs for the containers accessing the CNS file volume.
- An unbound ReadWriteMany persistent volume referring to a CNS file volume can be bound by a persistent volume claim created by any Kubernetes user under any namespace. Make sure that only authorized users have access to Kubernetes to avoid security issues.

Configuring the CSI Driver to Access vSAN File Service Clusters

Depending on the configuration, the CSI driver can provision file volumes on one or several vSAN clusters where the file service is enabled.

You can restrict access to only specific vSAN clusters where the file service is enabled. When deploying the Kubernetes cluster, configure the CSI driver with access to specific file service vSAN clusters. As a result, the CSI driver can provision the file volumes only on those vSAN clusters.

In the default configuration, the CSI driver uses any file service vSAN cluster available in vCenter Server for the file volume provisioning. The CSI driver does not verify which file service vSAN cluster is accessible while provisioning file volumes.

vSphere Cloud Native Storage Roles and Privileges

The CNS vSphere user must have specific privileges to perform operations related to Cloud Native Storage.

You can create several roles to assign sets of permissions on the objects that participate in the Cloud Native Storage environment.

Note These roles need to be created only for generic Kubernetes clusters. If you work in the vSphere with Tanzu environment, use the Workload Storage Manager role for storage operations.

For more information about roles and permissions in vSphere, and how to create a role, see the *vSphere Security* documentation.

Role Name	Privilege Name	Description	Required On
CNS-Datastore	Datastore > Low level file operations	Allows performing read, write, delete, and rename operations in the datastore browser.	Shared datastore where persistent volumes reside.
CNS-HOST-CONFIG-STORAGE	Host > Configuration > Storage partition configuration	Allows vSAN datastore management.	Required on a vSAN cluster with vSAN file service. Required for file volume only.
CNS-VM	Virtual machine > Change Configuration > Add existing disk	Allows adding an existing virtual disk to a virtual machine.	All cluster node VMs.
	Virtual Machine > Change Configuration > Add or remove device	Allows addition or removal of any non-disk device.	
CNS-SEARCH-AND-SPBM	CNS > Searchable	Allows storage administrator to see Cloud Native Storage UI.	Root vCenter Server.

Role Name	Privilege Name	Description	Required On
	VM storage policies > View VM storage policies	Allows viewing of defined storage policies.	
Read-only	Default role	Users with the Read Only role for an object are allowed to view the state of the object and details about the object. For example, users with this role can find the shared datastore accessible to all node VMs. For zone and topology-aware environments, all ancestors of node VMs, such as a host, cluster, and data center must have the Read-only role set for the vSphere user configured to use the CSI driver and CCM. This is required to allow reading tags and categories to prepare the nodes' topology.	All hosts where the nodes VMs reside Data center

Create a Storage Policy for Kubernetes in vSphere

The vSphere storage object that will back a Kubernetes containerized application should meet specific storage requirements. As a vSphere user, you create a VM storage policy based on the requirements given by the Kubernetes user.

The storage policy will be associated with the virtual disk or vSAN file share that back the Kubernetes container.

If you have multiple vCenter Server instances in your environment, create the VM storage policy on each instance. Use the same policy name across all instances.

If you want to create a storage policy for a datastore other than vSAN, see Creating and Managing vSphere VM Storage Policies. To use thick provisioning on a VMFS datastore, make sure to select an appropriate option in the **Volume allocation** drop-down menu when you define the policy.

For information about types of datastores that Cloud Native Storage supports and which datastores support thick provisioning, see vSphere Functionality Supported by vSphere Container Storage Plug-in in *Getting Started with VMware vSphere Container Storage Plug-in*.

Prerequisites

- The Kubernetes user identifies the Kubernetes cluster where the stateful containerized application will be deployed.

- The Kubernetes user collects storage requirements for the containerized application and communicates them to the vSphere user.
- Required privileges: **VM storage policies. Update** and **VM storage policies. View**.

Procedure

1 In the vSphere Client, open the **Create VM Storage Policy** wizard.

   a Click **Menu > Policies and Profiles**.

   b Under **Policies and Profiles**, click **VM Storage Policies**.

   c Click **Create**.

2 Enter the policy name and description, and click **Next**.

Option	Action
vCenter Server	Select the vCenter Server instance.
Name	Enter the name of the storage policy, for example `Space-Efficient`.
Description	Enter the description of the storage policy.

3 On the **Policy structure** page under Datastore-specific rules, select **Enable rules for vSAN storage** and click **Next**.

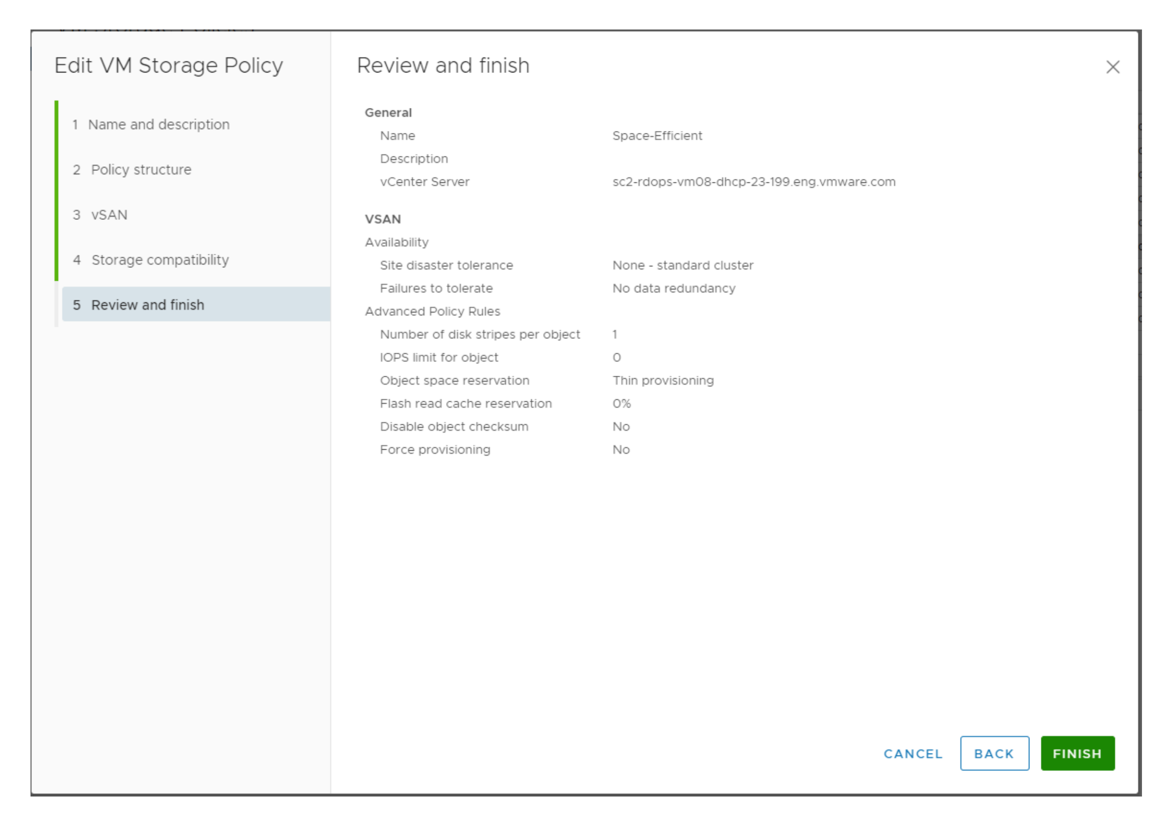
4 On the **vSAN** page, define the policy rule set and click **Next**.

   a On the **Availability** tab, define the **Site disaster tolerance** and **Failures to tolerate**.

   b On the **Advanced Policy Rules** tab, define advanced policy rules, such as number of disk stripes per object and flash read cache reservation.

5 On the **Storage compatibility** page, review the list of vSAN datastores that match this policy and click **Next**.

6 On the **Review and finish** page, review the policy settings, and click **Finish**.



What to do next

You can now inform the Kubernetes user of the storage policy name. The VM storage policy you created will be used as a part of storage class definition for dynamic volume provisioning.

Configure Kubernetes Cluster in vSphere Virtual Machines

On each node VM, enable the `disk.EnableUUID` parameter, so that the vSphere VMs can successfully mount the virtual disks.

Perform these steps for each of the VM nodes that participate in the cluster.

Prerequisites

- Create several VMs for your Kubernetes cluster. For the VM requirements, see Requirements of vSphere Cloud Native Storage.
- Required privilege: **Virtual machine. Configuration. Settings**.

Note To avoid errors and unpredictable behavior, do not take snapshots of CNS node VMs.

Procedure

1 In the vSphere Client, right-click the virtual machine and select **Edit Settings**.

2 Click the **VM Options** tab and expand the **Advanced** menu.

3 Click **Edit Configuration** next to Configuration Parameters.

4 Configure the **disk.EnableUUID** parameter.

If the parameter exists, make sure that its value is set to True. If the parameter is not present, add it and set its value to True.



Name	Value
disk.EnableUUID	True

Monitor Container Volumes Across Kubernetes Clusters in vSphere

After a stateful application gets deployed in Kubernetes, the volumes and their backing vSphere storage objects become visible in the vSphere Client. You can display and monitor the volumes and troubleshoot any potential storage issues.

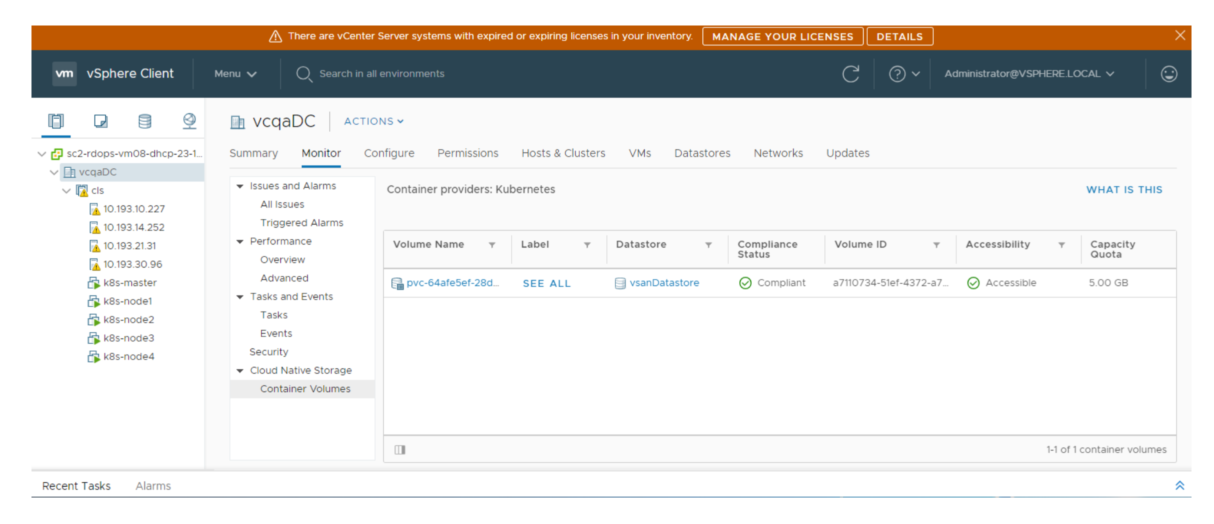
Note If you experience failures on the Kubernetes CNS server, the CNS objects in the vSphere Client might not display correctly until full synchronization takes place.

Procedure

1 Navigate to the vCenter Server instance, a data center, or a datastore.

2 Click the **Monitor** tab and click **Container Volumes** under **Cloud Native Storage**.

3 Observe the container volumes available in your environment and monitor their storage policy compliance status.

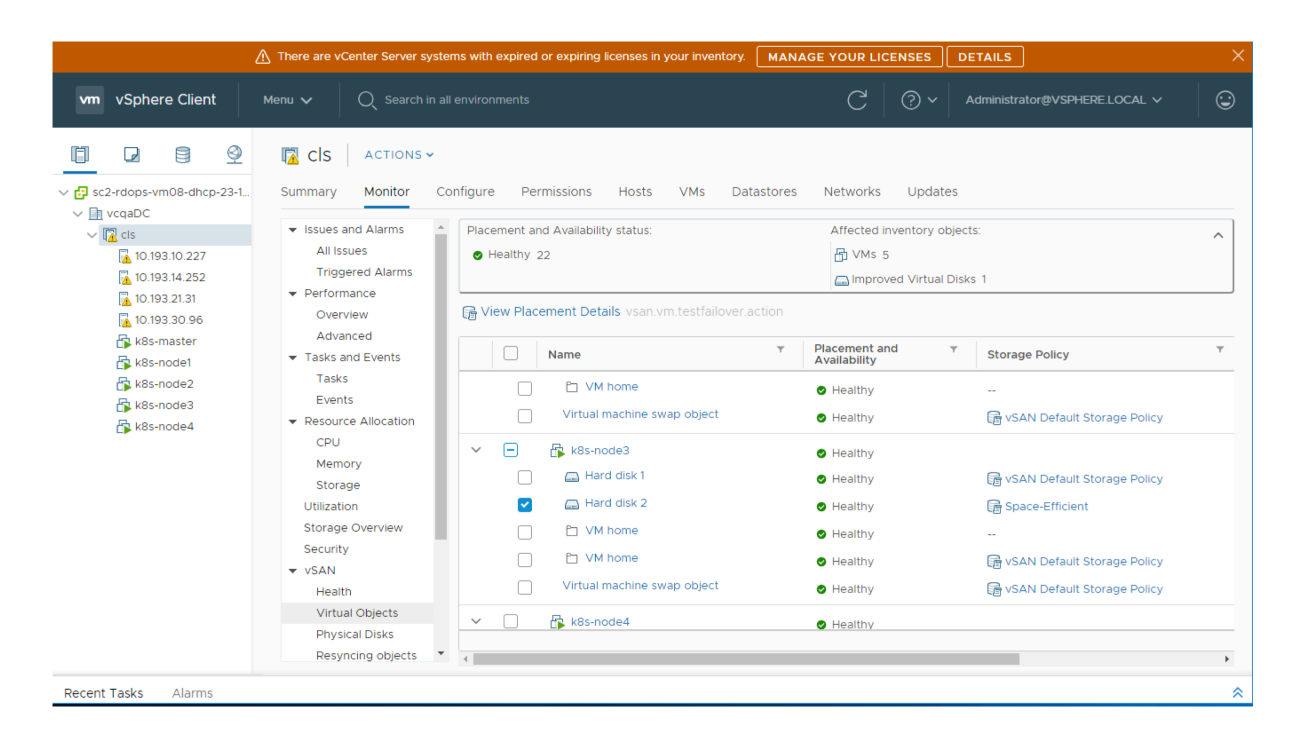


4 Click the **SEE ALL** link in the Label column to view additional details.

The details include the name of the PersistentVolumeClaim, StorageClass, and so on, and help you map the volume to the Kubernetes objects associated with it.

5 Click the link in the **Volume Name** column to review various components that back the volume and such details as placement, compliance, and storage policy.

Note The **Virtual Objects** screen is available only when the underlying datastore is vSAN.



Use Encryption with vSphere Cloud Native Storage

You can use the vSphere encryption technology to protect FCD virtual disks that back persistent volumes.

Using encryption in your vSphere environment requires some preparation, and includes setting up a trusted connection between vCenter Server and a key provider. vCenter Server can then retrieve keys from the key provider as needed. For information about components that participate in the vSphere encryption process, see vSphere Virtual Machine Encryption Components in the *vSphere Security* documentation.

Procedure

1 Set up the key provider in your vSphere environment.

   For information, see Set up the Key Management Server Cluster.

2 Encrypt all node VMs on the Kubernetes cluster.

   Use the vSphere Client to perform this step.

   a Navigate to a node VM.

   b From the right-click menu, select **VM Policies > Edit VM Storage Policies**.

   c From the **VM storage policy** drop-down menu, select **VM Encryption Policy** and click **OK**.

      To expedite the encryption process of the node VMs, you can encrypt only the VM home.

3 Create encrypted persistent volumes in the Kubernetes cluster with the vSphere CSI setup.

   a Create a StorageClass that references the VM Encryption Storage Policy.

      Use the following YAML file as an example.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: encryption
provisioner: csi.vsphere.vmware.com
parameters:
  storagePolicyName: "VM Encryption Policy"
  datastore: vsanDatastore
```

   b Use the PersistentVolumeClaim to provision the persistent volume.

      The PersistentVolumeClaim must include the name of the encryption storage class in the `storageClassName` field.

Migrating Container Volumes in vSphere

vSphere Cloud Native Storage supports migration of persistent container volumes between datastores.

Considerations and Limitations

When you perform the migration, keep in mind the following considerations:

- You can migrate only single block volumes.
- Make sure that original and target datastores are accessible to all nodes of the Kubernetes cluster. This applies to single zone deployments and to Kubernetes clusters deployed across multiple zones.
- If a volume is attached to a VM, the VM remains locked during volume relocation. If the volume you relocate is attached to a Kubernetes node VM, the system does not allow any other control operations on the VM. The operations include attaching or detaching other volumes to the VM, migrating the VM to a different datastore, changing any configuration of the VM, and so on.
- Cloud Native Storage does not support PMEM storage. Attempts to relocate a volume to a PMEM datastore fail.
- The target datastore must have sufficient free capacity to accommodate the migrated volume. If the free space of a datastore is less than the volume size, you cannot select the datastore for migration.
- The vSphere Client does not support automatic storage policy changes during the migration operation. However, the system doesn't prevent you from selecting a datastore with incompatible policy as a target datastore for migration.

- After the migration, you can locate the virtual disk that backs the volume in one of the following folders on a target datastore:
  - FCD directory on the datastore.
  - VM folder if the volume is attached to a VM.

For additional details, see the VMware Knowledge Base article 90607.

Migrate Container Volumes

You can use the vSphere Client to migrate the persistent container volumes.

Prerequisites

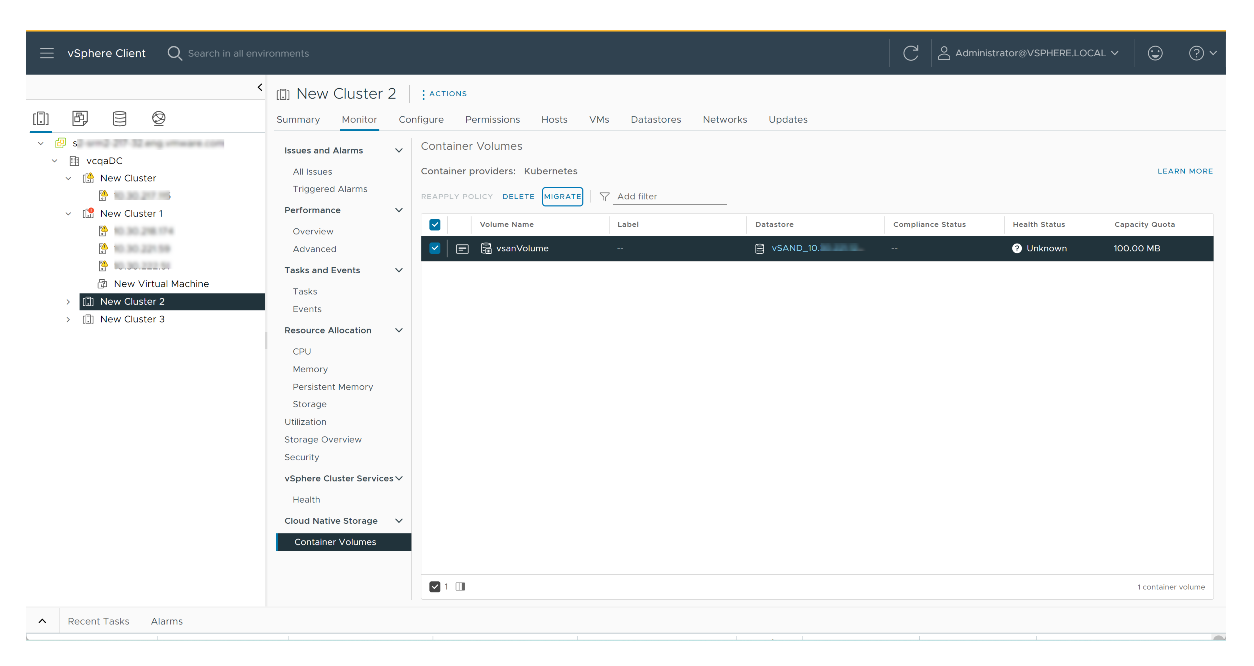
- Make sure that the target datastore has sufficient free capacity to accommodate the migrated volume.
- Make sure you have permissions for the target datastore.

Procedure

1 Navigate to the vCenter Server instance, a data center, a datastore, or a cluster.

2 Click the **Monitor** tab and click **Container Volumes** under **Cloud Native Storage**.

3 From the list of volumes, select a volume and click **Migrate**.

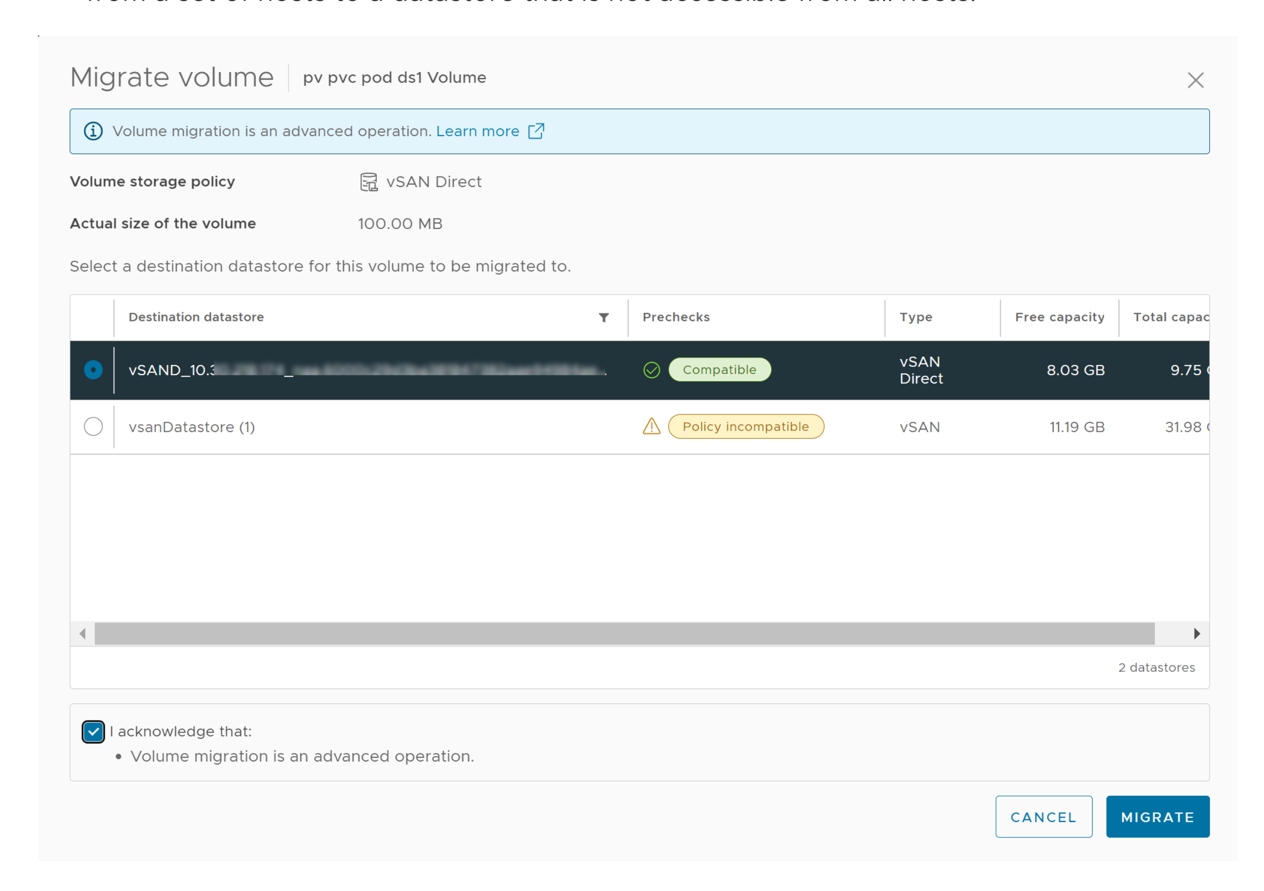


4 On the **Migrate volume** page, select the target datastore.

The target datastore might display a warning. You can still proceed with the migration operation when the warnings include the following:

- Policy incompatible. When the storage policy of the target datastore is not the same as the storage policy of the volume.

- Inaccessible volume. When the volume is migrated from a datastore that is accessible from a set of hosts to a datastore that is not accessible from all hosts.



5 Select the **Acknowledge** check box and click **Migrate** to proceed with the migration operation.